International Journal on Advances in Systems and Measurements

2011 vol. 4 nr. 1&2

The *International Journal on Advances in Systems and Measurements* is published by IARIA.
ISSN: 1942-261x
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Systems and Measurements
Volume 4, Number 1 & 2, 2011

- Jean-Henry Morin, University of Geneva - CUI, Switzerland
- Igor Podebrad, Commerzbank, Germany
- Leon Reznik, Rochester Institute of Technology, USA
- Chi Zhang, Juniper Networks, USA

Sensor Networks

- Steven Corroy, University of Aachen, Germany
- Mario Freire, University of Beira Interior, Portugal / IEEE Computer Society - Portugal Chapter
- Jianlin Guo, Mitsubishi Electric Research Laboratories America, USA
- Zhen Liu, Nokia Research – Palo Alto, USA
- Winston KG Seah, Institute for Infocomm Research (Member of A*STAR), Singapore
- Radosveta Sokkulu, Ege University - Izmir, Turkey
- Athanasios Vasilakos, University of Western Macedonia, Greece

Electronics

- Kenneth Blair Kent, University of New Brunswick, Canada
- Josu Etxaniz Maranon, Euskal Herriko Unibertsitatea/Universidad del Pais Vasco, Spain
- Mark Brian Josephs, London South Bank University, UK
- Michael Hubner, Universitaet Karlsruhe (TH), Germany
- Nor K. Noordin, Universiti Putra Malaysia, Malaysia
- Arnaldo Oliveira, Universidade de Aveiro, Portugal
- Candid Reig, University of Valencia, Spain
- Sofiene Tahar, Concordia University, Canada
- Felix Toran, European Space Agency/Centre Spatial de Toulouse, France
- Yousaf Zafar, Gwangju Institute of Science and Technology (GIST), Republic of Korea
- David Zammit-Mangion, University of Malta-Msida, Malta

Testing and Validation

- Cecilia Metra, DEIS-ARCES-University of Bologna, Italy
- Rajarajan Senguttuvan, Texas Instruments, USA
- Sergio Soares, Federal University of Pernambuco, Brazil
- Alin Stefanescu, University of Pitesti, Romania
- Massimo Tivoli, Universita degli Studi dell'Aquila, Italy

Simulations

- Tejas R. Gandhi, Virtua Health-Marlton, USA
- Ken Hawick, Massey University - Albany, New Zealand
- Robert de Souza, The Logistics Institute - Asia Pacific, Singapore
- Michael J. North, Argonne National Laboratory, USA

International Journal on Advances in Systems and Measurements
Volume 4, Numbers 1 & 2, 2011

CONTENTS

Increasing Measurability and Meaningfulness of Adaptive Security Monitoring by System Architectural Design and Mechanisms

Reijo M. Savola
VTT Technical Research Centre of Finland
Oulu, Finland
E-mail: Reijo.Savola@vtt.fi

Petri Heinonen
VTT Technical Research Centre of Finland
Oulu, Finland
E-mail: Petri.Heinonen@vtt.fi

***Abstract* — Decision-making in adaptive security management relies on sufficient and credible security evidence gathered from the system under investigation, expressed and interpreted in the form of metrics. If security measurability is not paid enough attention in advance, the availability and attainability of security evidence is often a major challenge. We propose and analyze practical and systematic security-measurability-enhancing mechanisms and system architectural design choices that enable and support adaptive and distributed security monitoring of software-intensive systems. The mechanisms are discussed in detail in the context of an adaptive, distributed message-oriented system. Examples of associated security monitoring techniques implemented in this environment are given. The study also discusses the feasibility of the proposed mechanisms. Security-measurability-enhancing mechanisms are crucial to the wider acceptance of security metrics, measurements, and associated tools and methods.**

Keywords* — *security monitoring; security metrics; adaptive security management; security measurability; message-oriented systems

I. INTRODUCTION

The constantly increasing complexity and connectedness of telecommunications and software-intensive systems, together with the greater number and variety of critical business applications operating in these systems, have heightened the need to implement carefully designed security mechanisms. As security threats and vulnerabilities, context of use, and protection needs change dynamically, adaptive security management and monitoring can provide effective and flexible security solutions. Security metrics can be used in resilient, self-protective, and self-healing systems to offer sufficient and credible security evidence for adaptive decision-making.

The US National Institute of Standards and Technology (NIST) published a roadmap report on directions in security metrics research [4]. This report argued that security metrics are an important factor in making sound decisions about various aspects of security, ranging from the design of security architectures and controls to the effectiveness and efficiency of security operations. The NIST report calls for practical and concrete measurement methods and *intrinsic security-measurability-enhancing mechanisms* within systems, motivating the research discussed in this study. Many security measurement challenges have their origin in the poor measurability support from the system under investigation. Security measurability can best be supported by designing enough support for it into the systems. Some of the mechanisms discussed in this study can help the designers with this task.

Adaptive security management should be capable of adapting to different use environments, contexts, and dynamic security threats. For instance, there can be different levels of authentication requirements: in some cases, strong authentication is needed and in others, multi-factor authentication mechanisms should be used.

The primary contribution of this study is the analysis of security-measurability-enhancing mechanisms for a distributed, adaptive security monitoring system, originally introduced in earlier work in [1]. Compared with the work in [1], this study provides more details and examples of the monitoring approach, and explains the mechanisms in greater detail. The mechanisms are investigated in the context of an example system, a distributed messaging system called Genetic Message Oriented Middleware (GEMOM) [2][3], which was developed in the European Commission's Framework Programme 7 project GEMOM (2008–2010). The GEMOM project developed a full-featured message broker, monitoring tool, and adaptive security management component, and it prototyped an intelligent fuzzing tool and anomaly detectors. The prototypes were validated in the following business-critical applications: banking transaction processing, financial data delivery, dynamic road management, collaborative business portal, and a dynamic linked exchange for procurement.

The rest of this paper is organized as follows. Section II briefly introduces the GEMOM system architecture and its security monitoring approach, while Section III discusses the use of security metrics in adaptive security monitoring. Section IV proposes and discusses novel technical mechanisms to enhance security measurability, and Section V analyzes the feasibility of the proposed mechanisms. Section VI discusses related work, and Section VII offers some concluding remarks and discusses future work.

II. SECURITY MONITORING APPROACH OF GEMOM

In the following, we discuss briefly the security monitoring approach of the GEMOM system. Although this study uses GEMOM as a reference system, the discussed solutions can be applied to many types of system architectures or communication mechanisms.

A. GEMOM Architecture

Message Oriented Middleware (MOM) solutions enable applications and systems to exchange messages with their communication parties without having to know the actual platform on which the application resides within the network. GEMOM is a scalable, robust, and resilient MOM system, the basic architecture of which is formed with GEMOM Nodes (G-Nodes) [5]. G-Nodes are either operational or managerial. Different configurations of G-Nodes can be used, depending on the current and future needs of the application or service.

One example GEMOM *subnet*, an operational system formed with G-Nodes, is depicted in Figure 1 [1]. The operational G-Nodes include Brokers, Publish Clients, Subscribe Clients, and Authentication and Authorization modules. Managerial G-Nodes include Adaptive Security Managers, Audit and Logging modules, Anomaly Monitors, Monitoring Tools (MTs), Security Measurement Managers, and Quality of Service (QoS) Managers. The managerial G-Nodes carry out runtime monitoring, control, and decisions in collaboration with the operational nodes [1]. The actual GEMOM system can consists of several subnets, or different types of nodes connected together, depending on the needs of the application, and resilience and performance issues.

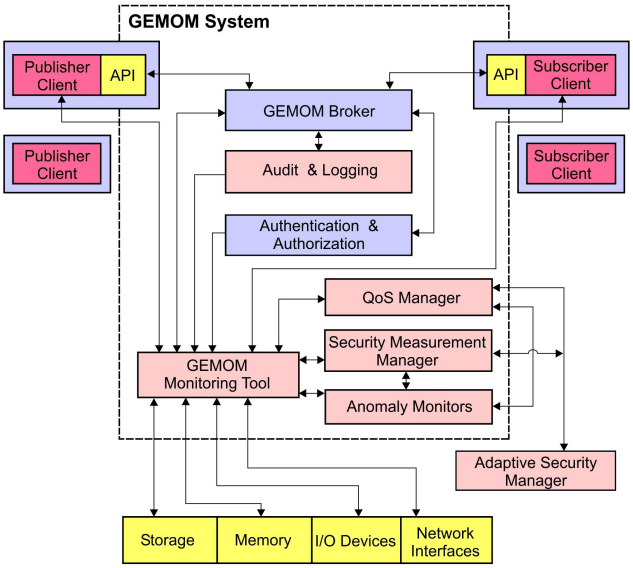

Figure 1. An example GEMOM subnet architecture [1]

Figure 1 also visualizes the connections to and from an MT [1]. The main entity responsible for adaptive security management in GEMOM, the Adaptive Security Manager node resides in the Overlay Manager and can be used to manage several subnets. In addition to the different types of nodes, the MT interfaces directly to platform resources, such as storage, memory, Input/Output (I/O) devices, and network interfaces. The Audit and Logging node provides internal functionality information at method or function level. For example, failures in methods and function calls, and user action logging can be monitored using this node.

B. Topics and Namespaces

GEMOM uses a publish/subscribe paradigm for message communication: authorized nodes can subscribe to *topics* and *namespaces* (see Figure 2) [1]. Publisher and Subscriber Clients have a core role in the content management approach of GEMOM. Communication architecture based on the publish/subscribe principle supports flexibility and scalability objectives well.

The topic contains published data in ⟨*key*, *value*⟩ format. The published data can be delivered to a Subscriber Client or another authorized G-Node, such as a node used in monitoring. Namespaces are mainly used for classification, with a namespace being a prefix for each topic. Namespaces are a higher level hierarchical construct compared with topics. For example, a measurement namespace can be reserved for measurement purposes in monitoring. Similar or associated topics belong to the same category, represented by the namespace. A topic can be shared by several Brokers or assigned to just one [1].

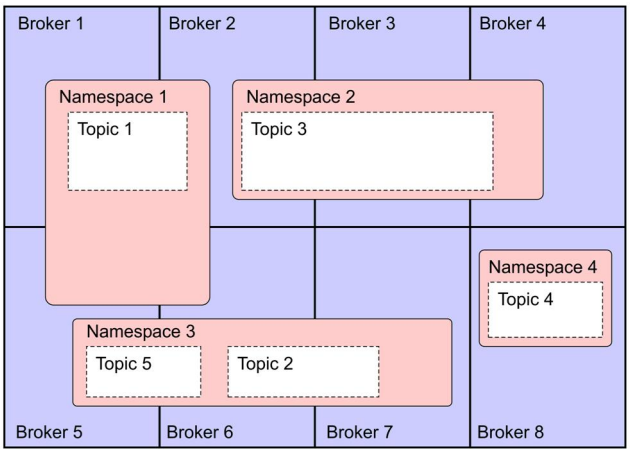

Figure 2. A visualization of topics and namespaces [1]

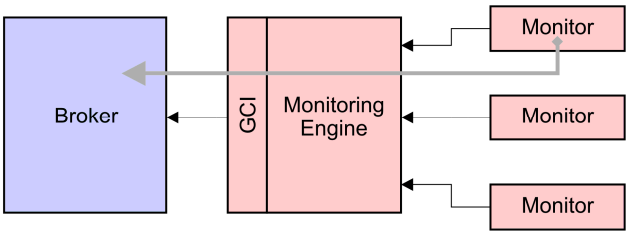

Figure 3. Communication between the Broker and the Monitoring Tool [1]

The actual physical locations of namespaces and topics are transparent to application users. A Subscriber can make a subscription to a topic or a namespace. The GEMOM Authentication and Authorization module manages the access rights to them. Namespace changes to a namespace

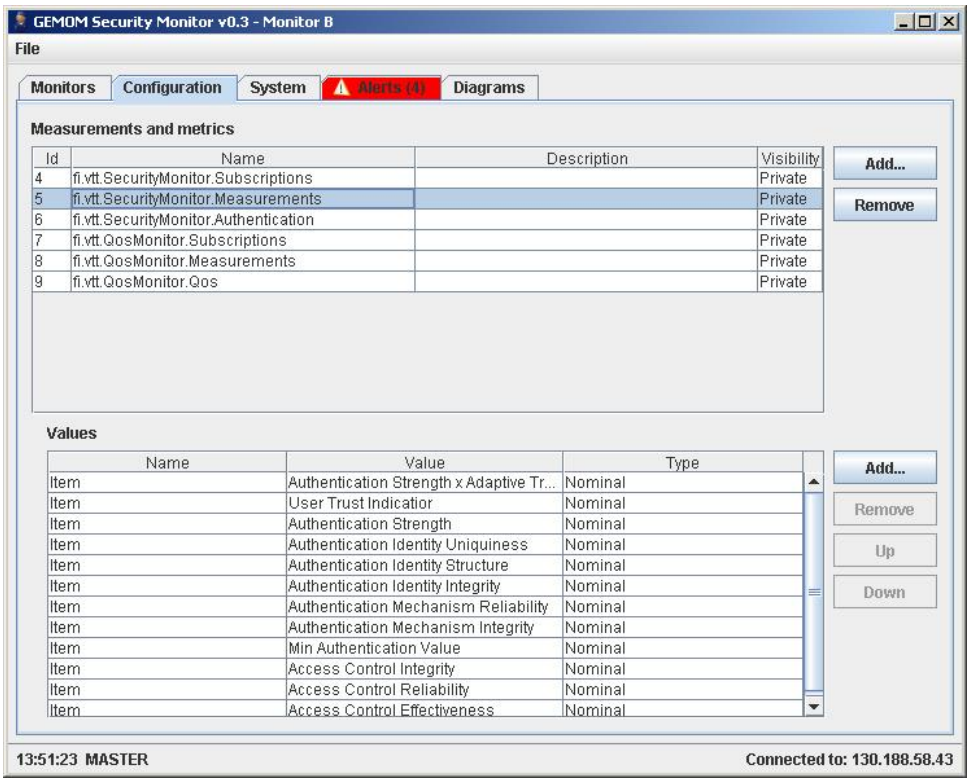

Figure 4. Configuration of authentication and authorization metrics in the MT

subscribed to by a Subscriber Client, such as deletions, name changes, or additions, or removals of topics, will continually be reported by the Broker to it. Topic changes (contents change, name changes, addition or removal of topic) to a subscribed topic are also reported [1]. The depth of reporting can be configured based on needs. Table I shows examples of namespaces and topics connected to the MT functionality.

TABLE I. EXAMPLES OF NAMESPACES AND TOPICS CONNECTED TO MT FUNCTIONALITY

Namespace, topic	Explanation
`/smon/modules, SecurityMonitor_x`	Under the `modules` namespace, there is a list of topics that contains module configurations for each security monitor.
`/smon/alerts/ SecurityMonitor_x, AnomalyMonitor`	`Alerts` namespace contains subnamespaces for each monitor. Subnamespaces contain topics for alerts sent from each Monitor module.
`/smon/metrics/ SecurityMonitor_x, Metric_y`	`Metrics` namespace contains subnamespaces for each monitor. Under them, there are metric configurations.

The GEMOM Broker is a core module of the whole system. It is a functional entity that consists of an application server, numerous plug-and-play objects, configuration files, and database objects [5].

C. Monitoring Tool and Monitor Modules

An MT controls the collection and further processing of security and QoS evidence and manages associated metrics and measurement databases. The main functional components of MT are the Monitor Engine and Monitor modules. The Monitor Engine implements Monitor Core software process functionality with the database and messaging service running in the background. The Monitor Engine does not carry out any monitoring itself but offers basic monitoring support services to the Monitor modules. The Monitor modules can either be pre-configured or added dynamically during the runtime operation [1].

In a GEMOM subnet, an MT is connected directly to the GEMOM Broker. The connection between the Monitor modules and Brokers is arranged via the GEMOM Client Interface (GCI) (see Figure 3) of the Monitoring Engine. GCI is a special interface component optimized for the GEMOM environment. The interface to other modules of the subnet is arranged differently: other modules use the GEMOM publish/subscribe mechanism to communicate and measure, publish and subscribe to relevant topics in a *measurement namespace* [5]. MTs use this mechanism to connect to Authentication and Authorization modules, QoS Managers, Anomaly Detector modules, and Security

Measurement Managers, as well as relevant used and free memory entities, storage (hard disks, memory sticks), network interfaces, and input/output devices (e.g., keyboard) [1]. In addition to logs and direct measurement results, the measurement data can include messages and metadata relevant for the measurement, as well as reports from associated security assurance tools.

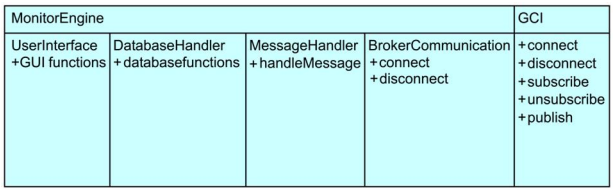

Figure 5. Sub-modules of Monitor Engine [1]

The monitoring system allows Monitor tools to be configured for different purposes, such as security, QoS, availability, and content monitoring. Only metrics, interfacing, and measurement spaces differ depending on the monitoring objectives. In the GEMOM environment, the QoS Monitor, Security Monitor and Anomaly Monitor tools have been implemented to be used for monitoring needs. Figure 4 shows a screenshot from the GEMOM MT, depicting the configuration management of the authentication and authorization metrics. The MT is in master mode. Alerts can be investigated by clicking the 'Alerts' tab during monitoring. Graphical visualization of monitoring results is obtained from the 'Diagrams' tab.

The MT Monitor Engine starts its operation as a Windows service. The sub-modules of the Monitor Engine and GCI interface are listed in Figure 5. After starting, the Monitor Engine will run in the background and be ready to establish a connection to a Broker using the BrokerCommunication sub-module by using the GCI interface. This sub-module also acts as a mediator for all publish and subscribe calls that originate from the MT [1].

Once the connection has been established, the Master MT switches to online mode. The first MT in the GEMOM subnet will obtain Master status. In the case of a machine crash or reboot, every MT is able to start up automatically and switch to online mode after the machine is up and becomes operational again. The MessageHandler sub-module handles messaging between the Monitor Engine and Monitor modules. The DatabaseHandler offers database services for Monitors. The UserInterface sub-module implements the user interface that is used for the configuration and management of MTs and the entire monitoring system. All events, statistics, status,

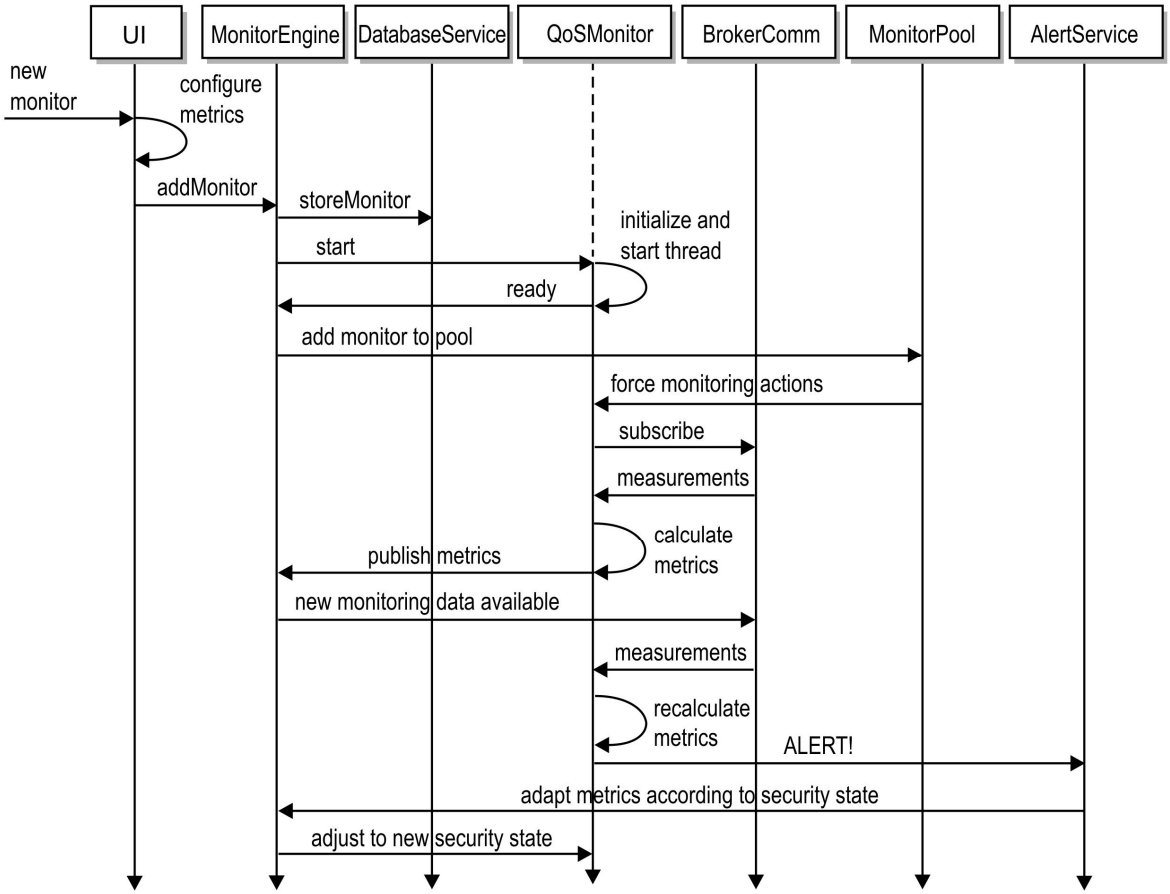

Figure 6. Adding a Monitor module to the system

functionality, and alarms are reported and visualized by the UserInterface sub-module. In addition, all information is available in special namespaces and topics [1]. Monitor modules that communicate with other MTs are connected to other Brokers and modules can be added to support multi-point monitoring needs. All distributed MTs have up-to-date monitoring information at their disposal. The MTs that reside close to the measurement points gather data from these points and make them available to other MTs.

D. Addition of New Monitor Modules

Security monitoring is typically carried out in co-operation with several Monitor modules, as completeness and meaningfulness of the measurements often require information from several system components and stakeholders.

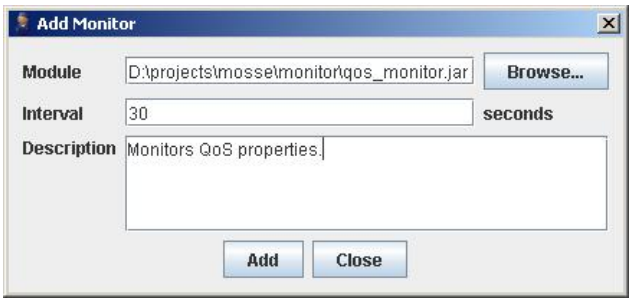

Figure 7. Adding new monitors is straigthforward in GEMOM. It is crucial for security-measurability to pay attention to ease of use

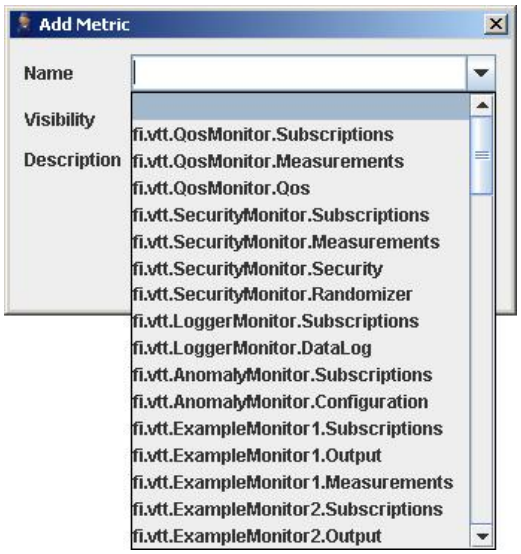

Figure 8. Adding metrics in a Monitor

Adding new Monitors is relatively straightforward; the main task in this context is to define appropriate decision-making algorithms and metrics. Figure 6 shows the functionality of adding a Monitor module to the system and Figure 7 a dialog box for end-user interaction during this operation. Once the metrics have been configured from the User Interface (UI), the Monitor can be saved and started. The Monitor initializes its own parameters and starts a thread. After this, it is a shared resource for the whole monitoring system. The Monitor carries out preliminary analysis and then receives measured data from the system. The measured data ('raw data') are used by the configured metrics collection and the metrics results are published for use by authorized subscribers, such as the Adaptive Security Manager component. If criteria for anomalies or other critical situations are met, alarms are raised. After this, all relevant Monitor metrics are adapted to the current situation. Adaptation can be carried out by, e.g., tightening selected requirements and criteria for new alarms, changing the frequency of measurements, or reducing the processing load.

E. Addition and Configuration of Metrics

Metrics can be added to the GEMOM Monitors with an easy-to-use user dialog box (see Figure 8). Metrics consist of logical expressions with either raw input data or results from other metrics. For each metric, the following parameters need to be configured: Metric ID, input data, output data, metric expression formula, threshold values, and timing. The following configuration presents an example of a Security Monitor metrics configuration:

```
fi.vtt.SecurityMonitor Metrics:
  fi.vtt.SecurityMonitor.Subscriptions

Topic@Namespace:130.188.58.43:7891@/_sys/clients
  fi.vtt.SecurityMonitor.Measurements
    Item:nAUTH
  fi.vtt.SecurityMonitor.Security
    MinimumAuthenticationStrength:0.3
    InitialNumber:0.7
    Lowest:0
    Highest:1
    Speed:0.05
```

According to the above configuration, `SecurityMonitor` subscribes to a topic that is updated by a Broker. It measures `nAUTH` (normalized authentication strength, *AS* in Eq. 1). If `nAUTH` drops below 0.3, `SecurityMonitor` will send an alarm. The following configuration is an example of QoS monitoring:

```
fi.vtt.QoSMonitor Metrics:
  fi.vtt.QoSMonitor.Subscriptions

Topic@Namespace:130.188.58.43:7891@/_sys/clients

Topic@Namespace:130.188.58.43:6148@/_sys/clients
  fi.vtt.QoSMonitor.Measurements
    Item:iCPU
    Item:aMEM
    Item:dBR
    Item:pBR
  fi.vtt.QoSMonitor.QoS
    MinimumAvailableMemory:500
    MinimumIdleCPU:50
    MaximumDataRate:1000
    MaximumPublishRate:10
```

With this configuration, `QoSMonitor` subscribes to the topics that are updated by Broker and GAgent and is able to read variables from them. It is configured to measure `iCPU` (CPU idle time, IT_{cpu} in Table IV), `aMEM` (available memory in Megabytes, *AM* in Table IV), `dBR` (data rate, bytes per second), and `pBR` (publication rate, publications per second). It also has a QoS metric that indicates when values are not acceptable. If a measured value is not in the acceptable range, `QoSMonitor` will send an alarm. The above configuration enables `QoSMonitor` to plot the measured values in a diagram.

III. Using Security Metrics for Adaptive Security Monitoring

The following section briefly discusses the security metrics that form the basis for security monitoring. It also provides an example of their use in adaptive security decision-making.

A. Development of Security Metrics in a Hierarchical Way

The term *security metrics* has become standard when referring to the security level, security performance, security indicators or security strength of a system under investigation – a technical system, product, service, or organization [6]. The general motivation for security measurements is the common argument that an activity cannot be managed well if it cannot be measured [7]. The above is particularly applicable to adaptive security management. Security solutions with varying strength levels are required in distributed networked systems such as GEMOM so that they can manage security in an adaptive manner according to the needs of varying situations like the context and dynamicity of security threats. Security metrics provide the means with which to score different solutions during the system's operation [7]. Measurements provide single-point-in-time views of specific, discrete factors, while metrics are derived by comparing two or more measurements taken over time with a predetermined baseline [8]. Security metrics can be used for decision support in assessment, monitoring, and prediction.

Our earlier work [7][9] analyzed the collection of security metrics heuristics developed to measure the correctness and effectiveness of the security-enforcing mechanisms of the GEMOM system. Security metrics have been developed for adaptive security, authentication, authorization, confidentiality, integrity, availability, and non-repudiation mechanisms. Extensive surveys of available security metrics can be found in [10][11][12]. The earlier mentioned research [7] introduced an iterative methodology for security metrics development that has been simplified here:

1. Carry out prioritized threat and vulnerability analysis of the system under investigation.
2. Use suitable security metrics taxonomies and/or ontologies to further plan the measurement objectives and metrics types.
3. Develop and prioritize security objectives.
4. Identify Basic Measurable Components (BMCs) from the security requirements using a decomposition approach. BMCs are leaf components of the decomposition that clearly manifest a measurable property of the system. Similarly, decompose the system architecture into components.
5. Define measurement architecture and evidence collection**.** Match the BMCs with the relevant system components with attainable measurable data.
6. Integrate metrics from other sources and select BMCs based on feasibility analysis.
7. Develop an appropriate balanced and detailed collection of metrics from the BMCs.

BMCs are identified by security objective decomposition [6][7]:

1. Identify successive components from each security requirement that *contribute to its security correctness, effectiveness and/or efficiency* [6], or another security property in question;
2. Examine the subordinate nodes in order to determine whether further decomposition is required. If so, repeat the process with the subordinate nodes as current goals, breaking them down into their essential components.
3. Terminate the decomposition process when none of the leaf nodes can be decomposed further or when further analysis of these components is no longer necessary.

Originally, the idea of security objective decomposition was proposed by Wang and Wulf [13]. Note that the mechanism of developing security metrics discussed above is one example of systematizing this task. There are so many other possible ways to develop metrics.

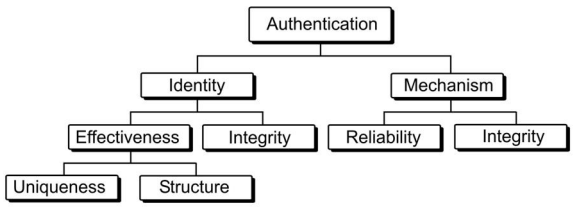

Figure 9. An example of authentication decomposition [13]

TABLE II. BMCs for Authentication [7]

Symbol	Basic Measurable Component
AIU	Authentication Identity Uniqueness
AIS	Authentication Identity Structure
AII	Authentication Identity Integrity
AMR	Authentication Mechanism Reliability
AMI	Authentication Mechanism Integrity

B. An Example of Authentication Decomposition

Figure 9 provides an example of high-level decomposition of main authentication objectives, while Table II shows the associated BMCs identified from this decomposition. The identity concept and authentication mechanism essentially contribute to the security strength of authentication. A more detailed explanation of the listed BMCs can be found in [7].

The metrics can be aggregated in the form of a weighted sum, resulting in Authentication Strength *AS* [7]:

$$AS = w_{AIU} \cdot \overline{AIU} + w_{AIS} \cdot \overline{AIS} + w_{AII} \cdot \overline{AII} + w_{AMR} \cdot \overline{AMR} + w_{AMI} \cdot \overline{AMI}, \quad (1)$$

to quote, "where w_x is the weighting factor of component x and '¯' denotes normalization and uniform scaling of the component metrics."

C. An Adaptive Authentication Management Example

The following example illustrates the use of the authentication metrics of Table II in adaptive security monitoring.

The GEMOM system requires a minimum Authentication Strength, min(AS_{usr}), for each user, *usr*. This value differs between the normal mode and the mode during suspicion of a Denial of Service (DoS) attack ('DoS alarm mode'). The conditions for min(*AS*) are [1]

$$\min(AS_{USR}): \quad AT_{usr} \cdot AS_{usr} \geq \theta_1 \wedge \quad \min(AIU_{usr}, AIS_{usr}, AII_{usr}, AMR_{usr}, AMI_{usr}) \geq \theta_2, \quad (2)$$

to quote, "where AT_{usr} is an adaptive trust indicator, θ_1 is the general Authentication Strength threshold and θ_2 is the threshold for each component metric. For instance, thresholds could be set as follows: during normal operation, $\theta_1 = 0.5$ and $\theta_2 = 0.2$, and during DoS alarm mode, $\theta_1 = 0.6$ and $\theta_2 = 0.3$." More details on parameters can be found in [7].

Below, we show a highly simplified scenario of how the authentication metrics discussed above can be used in an adaptive manner in GEMOM. The example consists of seven steps that represent the GEMOM security monitoring system in different authentication situations (or 'steps') [1]:

1. Authentication of the user *usr* authenticating himself/herself for the first time in an office environment using a smart card.
2. Identification of *usr* in an office environment using a user name/password pair. There are several weeks between Steps 1 and 2, and the value of the trust indicator has been increased.
3. Availability has dramatically decreased (increased delay and decreased QoS), possibly due to a DoS attack.
4. Identification of *usr* in the office environment using the user name/password pair. The Authentication Strength of *usr* falls under the threshold, causing an alarm.
5. Identification of *usr* in an office environment using an X.509 certificate. The Authentication Strength is now over the required threshold level and the authentication is successful.
6. Normal mode is resumed after the DoS attack mode. The adaptive trust indicator has now been increased, but *usr* has attempted to read a topic without rights to do so. Consequently, the adaptive trust indicator must be decreased by a certain amount.
7. Identification of *usr* in a home office environment, with a GEMOM smart card and user name/password pair in use. The authentication is successful.

TABLE III. STEPS 1–7 OF AN AUTHENTICATION MONITORING EXAMPLE [1]

Param.	**St. 1**	**St. 2**	**St. 3**	**St. 4**	**St. 5**	**St. 6**	**St. 7**
delay	0.2	0.2	0.9	0.9	0.9	0.3	0.2
AT_{usr}	0.5	0.7	0.2	0.2	0.3	0.6	0.4
QoS ind.	0.9	0.9	0.1	0.1	0.1	0.8	0.9
AIU_{usr}	0.7	0.6	0.6	0.6	0.8	0.7	0.6
AIS_{usr}	0.7	0.7	0.7	0.7	0.7	0.7	0.7
AII_{usr}	0.7	0.7	0.7	0.7	0.7	0.7	0.7
AMR_{usr}	0.4	0.2	0.2	0.2	0.4	0.3	0.6
AMI_{usr}	0.4	0.2	0.2	0.2	0.4	0.3	0.6

Figure 10 [1] shows a screenshot of the MT window that depicts the Authentication Strength in the above-mentioned steps. Note that the time scale shown in the screenshot does not correspond to the real timing between the steps. Real timing from step to step can be days or weeks. The threshold level is shown as a red line. If the Authentication Strength is below the threshold, the authentication process – controlled by the Adaptive Security Manager – will not continue until the strength moves above the threshold. This can be achieved using stronger authentication mechanisms, or the system alarm mode is over. The values of the core metrics associated with the above steps are shown in Table III. Note that the values are only informative and are not based on real measurements. All values have been normalized and scaled to the interval 0...1. Note that the correlation of different events, such as publish and subscribe information and meta-data, is required to distinguish between normal system peak loads and increased traffic due to a DoS or a Distributed DoS attack.

In practice, Authentication Strength is affected by a large number of dimensions that can be depicted by a hierarchy of sub-metrics for *AS*. The above example therefore only shows how, in principle, authentication monitoring based on metrics can be used. Moreover, all data cannot be obtained from the administration domain of the metrics users: data originating from the different stakeholders' part of the authentication process, such as Identity Provider, is needed.

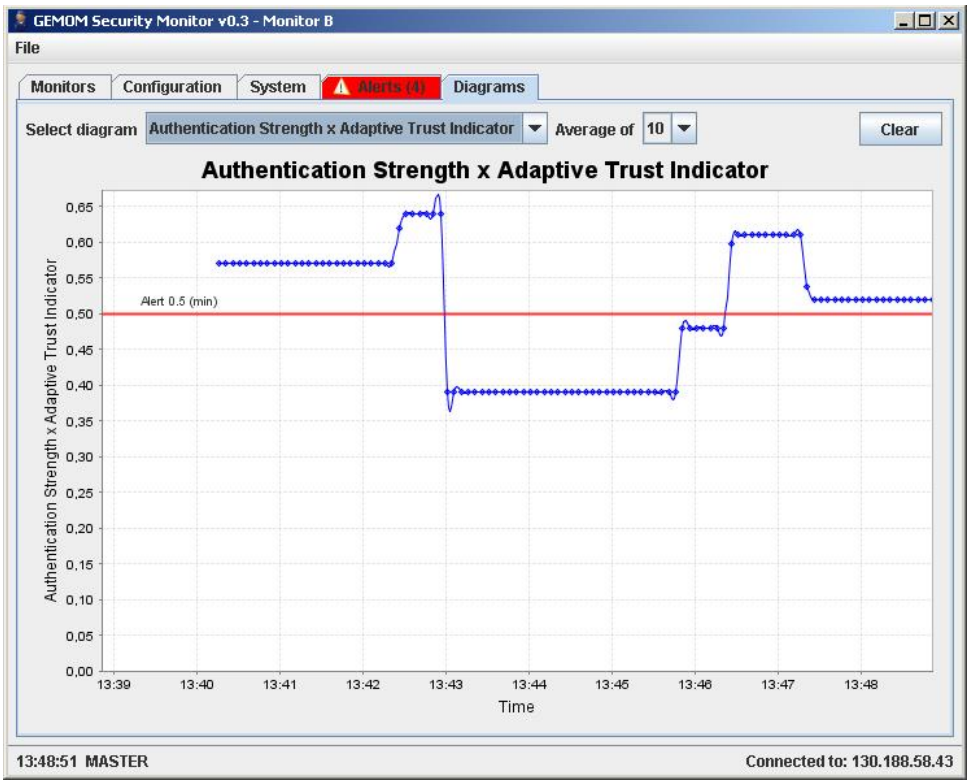

Figure 10. A screenshot of the GEMOM Monitoring Tool window depicting the seven steps of the authentication monitoring example [1]

IV. Proposed Security-Measurability-Enhancing Mechanisms

Although systematic and practical approaches to security monitoring based on metrics are generally desired, they are quite rare. One notable reason for this is that systems do not support security measurability very well. Measurability means that the metrics should be capable of having the dimensions, quantity, or capacity ascertained [14] in their measurement approaches [1].

The following section analyzes security-measurability-enhancing mechanisms that can be used to enable systematic and practical security monitoring in telecommunications and software-intensive systems. The mechanisms were introduced in our earlier work [1]. The mechanisms have been implemented in the research prototype of the GEMOM system [1].

A. *Flexible Communication Mechanism and/or Probing*

GEMOM nodes use the publish/subscribe mechanism to report their status and desired measurements to certain topics reserved for that purpose, while other authorized nodes can subscribe to this information. This mechanism is flexible, scalable, and increases the effectiveness of security measurements in the system. An example of how this mechanism can be used is presented in Figure 11 [1].

The performance of the actual system functions and communication can be measured and used for monitoring the design. For instance, a Broker can publish statistics of publish/subscribe activity, such as messages per second or kilobytes per second, or the delay between different nodes. An authorized node can subscribe to the associated topic or namespace. The monitoring system obtains measurement data from other nodes using the publish/subscribe mechanism. An indicator of monitoring delay, the relational 'distance' value of an MT part of the monitoring system, can be measured by comparing the delay data from Brokers. The delay data can be used for self-protection and resiliency management: if a Master MT crashes, an MT with the next highest priority automatically becomes a new Master MT. The prioritization can be partially based on the delay values of each MT.

If the publish/subscribe mechanism is not used, specialized *measurement probes* can be used to deliver measured data from the system components to the monitoring system. Different types of probes can be available or not available at different time instants and can be managed in a dynamic way. The probes should have a standardized language, yet be abstract, not related to any specific model. During monitoring, the measurement requirements have to be matched dynamically to the available and attainable probes that can deliver the required

measurement results [15]. The probes in use can reside in different parts of the system and in clients and servers.

The communication mechanism cannot fully solve the needs for gathering evidence from the platform resources, such as storage, memory, Input/Output (I/O) devices and network interfaces. Special measurement probes should be developed to manage this kind of evidence gathering.

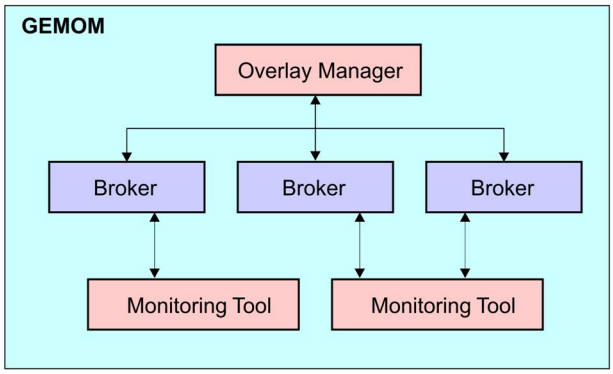

Figure 11. An example of using the publish/subscribe communication mechanism [1]

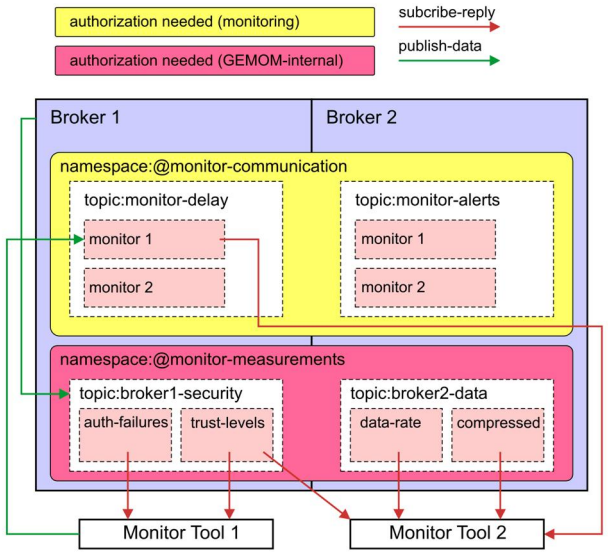

Figure 12. An example set-up of Monitoring Tools in an multi-point GEMOM monitoring environment [1]

It must be noted that the aggregation of individual security metrics results and measurements can be troublesome. Practical experience among industrial practitioners on aggregation has shown that the higher the abstraction level, the 'greener' the results tend to be, assuming a traffic lights approach ('red' meaning low level of security assurance or level, 'yellow' mediocre, and 'green' good) [15].

B. *Security Measurement Mirroring and Data Redundancy*

In general, GEMOM uses redundancy techniques to secure continued, uninterrupted operation in case of failures, overload, or a Denial-of-Service (DoS) attack [2]. As part of GEMOM's resilience management, redundant functionality, message feeds, and delivery paths will be maintained in the system in order to support switch-over to them in the event of a failure without information loss. Measurement data are also part of the redundancy functionality, and in addition to the nodes in operation, the measurement data reside in mirror nodes [1].

If a Broker crashes, valuable measured data about the failure event are available from its redundant counterpart, the *Mirror Broker* [1]. A Mirror Broker can be accessed separately for collection of data about the failure. Mirroring in security monitoring can be used for system security development purposes and learning from the 'what went wrong' evidence. A Mirror Broker and associated mirrored measurement data should use resources with no or only minor dependencies to the main Broker and measurements. If there are too many dependencies between the brokers, a failure in the main Broker can affect the Mirror Broker.

The effectiveness of mirroring can be increased by adding more than one Mirror Broker and/or mirrored data resources. In general, if the system contains N copies of the data, the resulting *redundancy level* is $N - 1$. The more mirrored functionality and data, the more processing resources are needed to carry out copying functionality associated with the mirroring procedure. Resources can be saved if smart mirroring procedures are used. For instance, mirroring can be done in a prioritized way in which most of the essential functionalities and/or data are kept up-to-date more frequently than less essential ones. Moreover, snapshots, raw presentations of the data, can be used.

C. *Multi-Point Monitoring*

A GEMOM subnet includes an Overlay Manager, n Brokers, and k MTs. An MT can monitor up to n Brokers (see Figure 12 [1]), which enables *multi-point monitoring*. In this subnet, one MT acts as a master and the other MTs act as slaves. The Master MT is responsible for synchronizing communication and data exchange between different MTs. While the status information of each MT is available for every MT, only the Master MT is allowed to manage others [1].

Figure 13 shows a GEMOM MT screenshot of the management of a Master and a Slave Monitor.

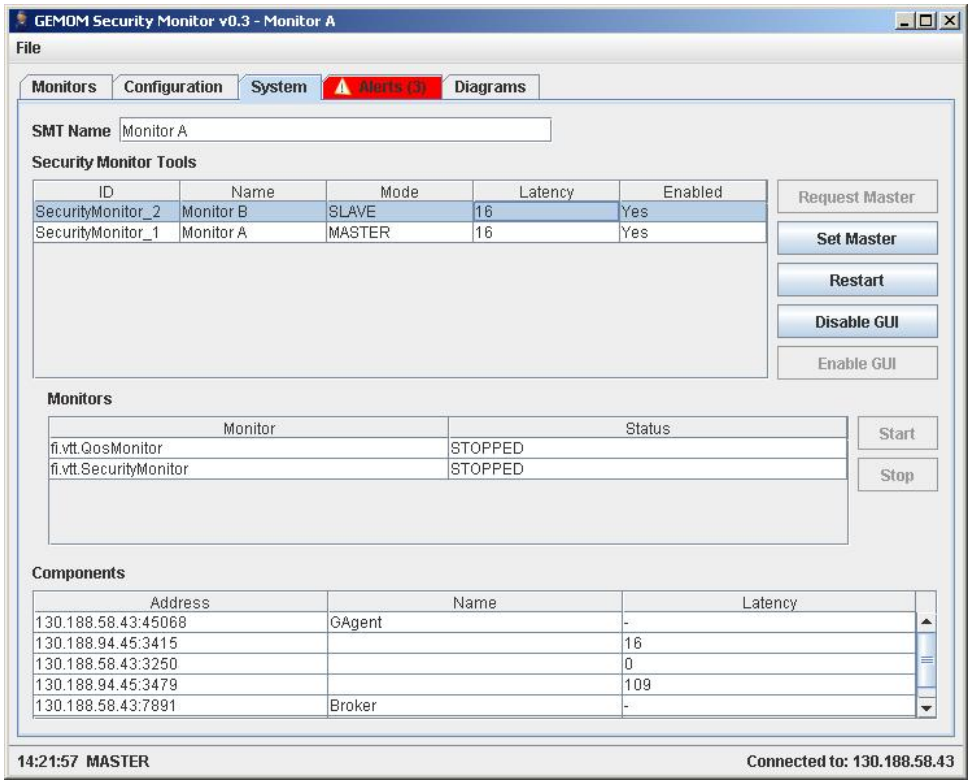

Figure 13. An MT window screenshot displaying a simple monitoring set-up for a Master Monitor with one Slave Monitor

In security-related measurements, multi-point monitoring is needed in most cases. Security issues in different parts of the overall system under investigation (e.g., a technical system, service, product) contribute to the overall security performance. Moreover, security evidence from stakeholders residing outside the administration scope of the system can be needed. For instance, the Identity Provider's identity establishment and management processes, and the client-server authentication protocols interfacing to it affect the Authentication Strength. Measured evidence is not always available and attainable from outside the scope of the system administration. In these cases, available or assessed evidence information can be configured directly to the Monitor modules. Such evidence is usually relatively static, and does not need to be updated frequently.

D. Auto-Recovery on Error

The MTs save their state information for later use in the case of a system crash due to a failure or attack to support flexible investigation of the situation. When the MT is rebooted, it activates auto-recovery functionality with the error mode and automatically initiates connection to the system. In this kind of situation, the crashed MT will always operate as a slave, because the error situation should be investigated before resuming normal operation, even if the crashed MT was a Master MT before the failure. If a new Master MT is started, it can easily disable the crashed and rebooted MT [1].

Auto-recovery functionality can be automated in different ways: (i) no auto-recovery, (ii) attempt auto-recovery for *N* times or *M* minutes, or (iii) attempt auto-recovery indefinitely. From a security monitoring perspective, the automation depends on the nature of the failure of the attack. It is important, especially after an attack situation, to investigate the situation and validate the monitoring configuration before, during, and after the attack in order to ensure that every MT plays its role as fairly as possible. Root-cause analysis techniques can be used to investigate the attack situation in detail based on the evidence provided by the MT.

Below is an example of an internal monitoring tool messaging and synchronization topic structure. The `/smon` namespace has two topics: `Watchdog` and `Control`. The `Watchdog` topic is updated by the Master MT and it is read by all Slave MTs. They should reply with the watchdog time to `/smon/smt` namespace under the `Alive` topic. If any Slave MT does not update its watchdog time, it is considered to be unavailable. The `Hierarchy` topic shows which monitor is playing the master role. In case the Master MT crashes, other monitors decide which will be the new master. The new master takes over the master role and starts controlling the

watchdog timer and the control messaging. Control messages are sent via the `Control` topic in the `/smon` namespace.

The `/smon/modules` namespace is used to inform other MTs of the kind of modules that are running in each monitor. The `/smon/metrics` namespace shows the actual metric configuration of all monitor modules.

```
/smon

  T: Watchdog
  Time=289

  T: Control
  from=SecurityMonitor_xx
  to=ALL|SecurityMonitor_xx
  time=12334214
  cmd=MASTER_MODE
      REQUEST_MASTER_MODE
      RESTART:<delay>
      ENABLE_GUI:true|false
      START_MONITOR:n@@xx.xxx.XxxXxxx
      STOP_MONITOR:m@@xx.xxx.XxxXxxx
      LASTID:xx


/smon/modules

    T: SecurityMonitor_0
    1@@fi.vtt.QoSMonitor=RUNNING
    2@@fi.vtt.AnomalyMonitor=STOPPED
    CurrentConfig=1,2

    T: SecurityMonitor_1
    1@@fi.vtt.ActivityMonitor=RUNNING
    CurrentConfig=1

    T: SecurityMonitor_2


/smon/smt

    T: Alive
    SecurityMonitor_0=289
    SecurityMonitor_1=289
    SecurityMonitor_2=289

    T: Names
    SecurityMonitor_0=Machine in Room E272

    T: Hierarchy
    SecurityMonitor_0=Master
    SecurityMonitor_1=Slave

    T: Latency
    SecurityMonitor_0=14
    SecurityMonitor_1=37


/smon/measurements

    /aMEM

      T: <ip>:<port>
      <timestamp_long>=<aMEM_average>

    /iCPU

      T: <ip>:<port>
      <timestamp_long>=<iCPU_average>


/smon/alerts

    /SecurityMonitor_0

      T: 1@@fi.vtt.QoSMonitor
      From=SecurityMonitor_0
      Discriminator=aMEM
      Status=NEW|OLD|ACK|DEL
      OldStatus=NEW|OLD|ACK
      Time=142412455
      Severity=CRITICAL|MAJOR|MINOR|WARNING
      Text=Something went terribly wrong!

      T: 2@@fi.vtt.ActivityMonitor

    /SecurityMonitor_1

      T: 1@@fi.vtt.QoSMonitor


/smon/metrics

    /SecurityMonitor_0

      T: fi.vtt.QoSMonitor.Broker
      MinimumAvailableMemory=335
      Type=Decimal
      Visibility=Public
```

E. Integrity, Availability and Configuration Correctness Checks

The integrity and availability metrics, part of the security metrics collection of GEMOM, are based on the results from the integrity and availability check functionality built into selected critical parts of the system. Integrity and availability checks are carried out by special program code constructs and algorithms at run-time and in connection with software security assurance activities (such as testing and analysis). In addition to code constructs, tools such as Tripwire [16] can be used for periodical and triggered integrity checks of files and data. The reports and potential alarms of integrity checks carried out by integrity check code constructs and tools are visible in the MT, and this evidence is used as part of the integrity metrics.

The integrity and availability checks address typical software weakness and vulnerability problems. The following widely known checks increase the security quality and can be used to support the security measurability [1]:

1. Validation of input data in all relevant interfaces. It is possible to prevent most injection attacks (such as Cross-Site Scripting, Structure Query Language SQL injection, and null injection) with by proper input validation.
2. Buffer and memory overflow checks
3. Storage and database checks
4. Error recovery, self-protection and resiliency checks

Configuration correctness is one of the main aims of integrity checks, along with high quality of software and a lack of critical vulnerabilities. Metrics depicting configuration correctness are based on design and implementation requirements, reference standards, and best practice information. The wrong configuration and deployment of security controls can result in severe security problems. Moreover, wrong system configuration in other parts than those directly linked to security functionalities can potentially turn into security problems.

Further checks should be developed, such as focusing on the issues pointed out by the results of the threat and vulnerability analysis of the SuI, and available public vulnerability information, such as OWASP [17] classifications and applications of them, e.g., [18].

It must be noted that it is possible to develop a wide collection of integrity and availability metrics, and the checks can degrade the processing performance greatly. Prioritization of the results of threat and vulnerability analysis is therefore important.

F. State of Security Framework

The timing of security-related measurements needs to be managed carefully. Risk assessment is essentially a prediction of security risks that can cause problems in the future. When the security controls that are implemented and deployed in the system are based on the output from risk assessment, it is important to handle history information, current measurements and future predictions separately. The MTs and the Adaptive Security Management functionality in the Overlay Manager use the State of Security (SoS) framework [5] (see Figure 14 [1]). This concept can be seen as a security-measurability-enhancing mechanism, implementing a timing framework for security measurements.

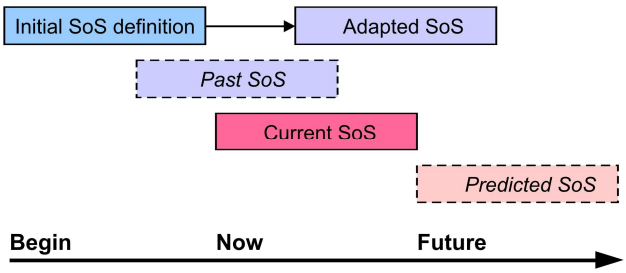

Figure 14. The high-level State of Security concept in GEMOM [1]

SoS is a time-dependent estimate of the system's security level and performance based on an appropriate integrated and aggregated balanced collection of security metrics. In short, SoS describes the high-level *security health index* of the operational GEMOM system. SoS contains an aggregated value of individual metrics components. The estimate is initially offered by the MT and whenever it is triggered. Metrics with different time scale properties (lagging, coincident, or leading) are used depending on the situation [1].

Note that valuable information is always lost in the aggregation process of metrics results. A graphical representation of metrics in the MT can therefore be used when investigating the SoS in more detail.

There are five different phases in the estimation of SoS [1]:

1. The *initial SoS* is calculated based on the collection of initial values of security metrics (lagging and coincident metrics). Weights are associated with different component metrics in order to indicate their relative importance, based on the results of risk-driven prioritization. In practice, a 'close to correct' weight assignment is used, as analytical results are often unavailable [19]. Moreover, practice has shown that the aggregated results tend to show results that are too optimistic compared with reality. The initial weights are assigned based on up-to-date threat, vulnerability, trust, and reputation knowledge. The metrics components that need to be balanced in the collection are adaptive security, authentication, authorization, confidentiality, integrity, availability, and non-repudiation.
2. The *current SoS* is based on a coincident metrics calculation whenever it is triggered by a timer, an attack alarm, an anomaly alarm, or a manual request. The weighting is adjusted based on updated data on threats, vulnerabilities, the context, or other relevant parameters.
3. Past and current SoSs can be *compared* in the monitoring system in order to identify trends and potential fault situations. Trend analysis is an important application of security monitoring results.
4. The SoS can be *adapted* according to decisions made by the Adaptive Security Management functionality in the Overlay Manager. The adapted SoS is the result of actions carried out by the Overlay Manager.
5. The *predicted SoS* is based on leading metrics to support proactive estimation. A comparison of past SoS can be used to identify trends affecting the prediction. The predicted SoS can be fed as input to the threat and vulnerability analysis.

Figure 15 depicts the general process from the Current SoS to the Adapted SoS. After the process, the old Current SoS becomes a Past SoS and the Adapted SoS becomes the Current SoS. The trigger mentioned in the figure can be an alarm, incident, or some other type of security effectiveness information. Trends and changes in the security risks affect the updates of metrics and their calculation.

G. Measurability Support from Building Blocks of Security

The architectural design of the system greatly affects the

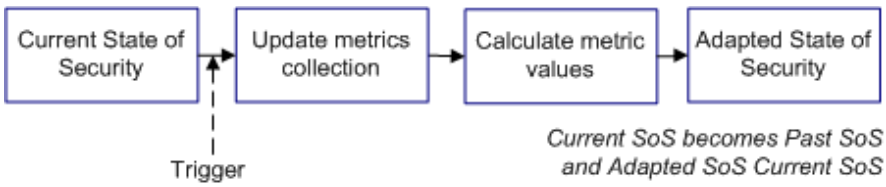

Figure 15. From Current SoS to Adapted SoS.

security measurability of the system during its operation. It is important that the various architectural security building blocks are designed to support seamless co-operation between the different parts of the operational system and its connections. In communication architectures, nodes or modules should be designed in such a way that it is possible to obtain enough data about their operational state in an authorized manner. Measurement probe architecture and the management of probes should be designed hand-in-hand with the actual system architecture. Stage 5 (design of measurement architecture), mentioned in the security metrics development approach discussed above, is tightly connected to the actual architecture of the system.

Several architectural components in GEMOM have been built in such a way that they exhibit internal properties and functionalities that support security measurements: in other words, they can be considered to incorporate *intrinsically security-measurable* [4] or security-measurability-enhancing constructs. These constructs increase the credibility, applicability, and sufficiency of monitoring approaches based on security metrics [1].

In addition to the architectural building blocks, systematic and automatic information exchange in GEMOM has been designed between the different stages of the security metrics development process: threat and vulnerability analysis, security requirements definition, decomposition of requirements, and detailed metrics definition [1].

H. Use of Shared Metrics and Measurement Repositories and Enumerations

Enumerations classify fundamental entities and concepts relevant to security, and repositories allow common, standardized content to be used and shared. Shared security metrics and measurement repositories support the development of credible security metrics and help to focus and prioritize efforts. In the near future, several novel shared security metrics and associated data repositories will be available. Examples of these include the Common Vulnerability Scoring System (CVSS) [20] and the associated baseline security data enumeration Common Vulnerabilities and Exposures (CVE) [21], both of which are part of the Security Content Automation Protocol (SCAP) [22]. Martin discussed their use in [23], along with how they integrate with different stages in the secure system development life cycle. In addition, the Web Application Security Consortium (WASC) and the SANS Institute maintain popular threat and vulnerability classification collections, found in [24] and [25], respectively [1]. Table IV recaptures examples of security enumerations, languages, and repositories [26].

I. Reuse of Available Metrics and Measures Relevant to Security

In many cases, parameters that were originally developed for other measurement purposes can be applied, at least partly, to security measurements. The following section provides some examples of these. Taking into account metrics that are readily available in the system is part of the action to increase security measurability. The available and attainable metrics relevant to security can be matched with the needs of security evidence.

TABLE IV. EXAMPLES OF SECURITY ENUMERATIONS AND REPOSITORIES [26]

Name	Explanation
Common Vulnerabilities and Exposures (CVE)	A standard enumeration of identifiers for known vulnerabilities
Common Weakness Enumeration (CWE)	A standard enumeration of identifiers for software weaknesses
Common Attack Pattern Enumeration and Classification (CAPEC)	A standard enumeration for identifiers for attacks
Common Configuration Enumeration (CCE)	A standard enumeration for identifiers for configuration
Common Platform Enumeration (CPE)	A standard enumeration for platforms, operating systems, and application packages
SANS Top-20	Top critical vulnerabilities list by the SANS Institute
OWASP Top-10	Top critical vulnerabilities list by OWASP
WASC Threat Classification	Threat classification by the WASC consortium
U.S. National Vulnerability Database (NVD)	A U.S. vulnerability database using, e.g., CVE, CCE, SCAP, and US-CERT data
Red Hat Repository	Patch definitiions for Red Hat Errata security advisories
OVAL Repository	OVAL vulnerability, compliance, inventory and patch definitions
Center for Internet Securtiy (CIS)	Security configuration benchmarks
U.S. Department of Defence Computer Emergency Response Team (DoD-CERT)	Information assurance vulnerability alerts ans security implementation guides
U.S. National Security Agency (NSA)	NSA security guides

TABLE V. MACHINE -RELATED COUNTERS [5]

Symbol	Counter
IT_{cpu}, PT_{cpu}	CPU idle time, CPU processing time
AM	Available memory
PA	Paging activity
B_{util}, B_{max}	Bandwidth usage, maximum bandwidth
L_{m2m}, V_{m2m}	Latency, visibility between two machines

TABLE VI. PUBLISH/SUBSCRIBE ACTIVITY-RELATED COUNTERS [5]

Symbol	Counter
PPN, PPT	Publications per namespace, topic
PPB, PPC	Publications per Broker, Client
MPB, MPC	Messages per Broker (publications and subscriptions), Client
DPB, DPC	Amount of data per Broker, Client
PBPC	Number of protocol breaches per Client
N_{ns}, N_{top}	Number of namespaces, topics

TABLE VII. SOURCES OF SECURITY METRICS IN GEMOM [5]

Symbol	1	2	3	4	5	6	7
Countermeasure mechanism performance metrics (by requirement decomposition)	×	×	×	×	×	×	×
Metrics based on anomaly and misuse models (attacker-oriented weakness metrics)		×	×				
Cryptographic strength metrics		×		×	×		×
Availability metrics based on QoS application performance metrics						×	
Trust metrics	×						
Reputation metrics	×						
Software quality metrics	×	×	×	×	×	×	×
Vulnerability metrics	×	×	×	×	×	×	×
System endemic security-relevant metrics						×	

QoS evidence can often be used for security-related availability measurement. Different types of availability threats usually have a high impact in telecommunication systems: the system, or a core part of it, is in its most vulnerable state during availability attacks, such as Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. Moreover, the actual execution of the application often crashes due to these attacks. An attacker can execute his/her strategy and achieve his/her goals by exploiting the high vulnerability time window, which can potentially also cause other security threats. The intruder can even seize the system using this strategy [27]. Until recently, QoS and security metrics have lived in separate worlds. Some security attacks affect application performance, and the most important objective of QoS management is to ensure application performance. The GEMOM monitoring approach uses metrics for security availability and application performance measurement [6]. Other availability parameters that can usually be obtained from telecommunication systems are delay, delay variation, and packet loss rate [27] [1].

Machine-related and functionality-related counters can be used in security measurements. Table V [5] lists some general machine-related counters that are available in the GEMOM system. Similar counters are available in most systems. Aggregated information from several counters can be used to identify system situations that could potentially be a threat to security. Table VI [5] lists some publish/subscribe activity-related counters that can be used when monitoring abnormal situations. Table VII [5] summarizes the origin of security metrics and other security-relevant metrics in GEMOM. The numbers in the columns represent the security-enforcing mechanisms (1 = adaptive security, 2 = authentication, 3 = authorization, 4 = confidentiality, 5 = integrity, 6 = availability and 7 = non-repudiation). '×' in the relevant box indicates the use of the security metrics in GEMOM [1]. Note that in another types of systems, different types of security metrics can be used, and their origin can be different.

J. Secure Coding – Security-Relevant Software Quality

Software quality affects the resulting security level of the system directly and indirectly. Software quality can be increased by the application of secure coding principles. In addition to input validation, secure coding principles include, for example [28], the defense-in-depth principle, the principle of least privilege, aiming for the simplest system possible, default access denial, data sanitization, effective quality assurance, and use of a secure coding standard (best practice).

It must be noted that some of the secure coding principles seem to decrease the security measurability. For example, default access denial, the principle of least privilege, and data sanitization may have this kind of effect due to the reduced amount of available information. Proper authentication and authorization should be implemented in order to allow the security measurement activity to obtain all relevant data.

Security metrics that address the enforcement of security coding principles are part of the balanced security metrics collection. Knowledge about the principles used increases the security measurability of the system.

V. IMPACT ON FEASIBILITY OF SECURITY MONITORING

The feasibility of security measurement and associated monitoring is at the core of its success. Our earlier work investigated the criteria for the quality and feasibility of using security metrics in software-intensive systems [29] and the core requirements for a secure and adaptive distributed monitoring approach [30]. The following section discusses the impact that the mechanisms introduced in this study have on the feasibility of security monitoring based on security metrics. The metrics criteria of [28] and the monitoring requirements of [30] are deployed as feasibility objectives [1].

The criteria in [29] are classified into three levels, each of which incorporates six criteria. The levels emphasize the credibility of security metrics, their applicability for use together with the measurement approach, their sufficiency for their intended use, and the completeness of the metrics collection. The criteria were originally developed for security metrics. The applicability of the security metrics and measurement approaches, such as monitoring and applying them to the final use environment, is crucial however. Security metrics should be designed 'hand-in-hand' with the measurement approach: metrics cannot be used without measurements, and measurements are useless unless they are interpreted [29].

Table VIII [1] summarizes the feasibility criteria discussed in [29] and provides examples of how they are supported by the security-measurability-enhancing mechanisms of the GEMOM monitoring approach. In the table, FC denotes flexible communication, SMM denotes security measurement mirroring and data redundancy, ARE indicates auto-recovery on error, MPM is multi-point monitoring, IAC represents integrity and availability checks, TF is the timing framework, BBS is the measurability support of building blocks of security, SMMR is the use of

shared metrics and measurement repositories, RAM is the reuse of available metrics and measurements relevant to security, and SC indicates secure coding. In general, the development of security metrics is challenging and metrics that meet all the feasibility criteria are extremely rare [29].

TABLE VIII. FEASIBILITY CRITERIA OF [29] AND DISCUSSED MECHANISMS SUPPORTING THEM [1]

Criterion	Supporting Mechanism(s)
Correctness	IAC, SC
Granularity	All
Objectivity and unbiasedness	TF, SMMR, RAM
Controllability	All
Time-dependability	MPM, ARE, TF
Comparability	FC, MPM, ARE, IAC, SMMR, SC
Measurability	All
Attainability, availability, easiness	All
Reproducibility, repeatability, scale reliability	All
Cost effectiveness	FC, BBS, SMMR, RAM
Scalability and portability	FC, SMM, MPM, BBS, RAM, SC
Non-intrusiveness	FC, SMM, MPM, ARE, IAC, BBS, RAM, SC
Meaningfulness	MPM, TF, BBS, SMMR, RAM
Effectiveness	MPM, ARE, TF, BBS, SMMR, RAM
Efficiency	FC, SMM, MPM, BBS, RAM, SC
Representativeness and contextual specificity	TF, BBS, SMMR, RAM
Clarity and succinctness	MPM, IAC, BBS
Ability to show progression	TF, SMMR
Completeness	FC, TF, BBS, SMMR, RAM

TABLE IX. ADDITIONAL MONITORING REQUIREMENTS OF [30] AND DISCUSSED MECHANISMS THAT SUPPORT THEM [1]

Requirement	Supporting Mechanism(s)
Security	All
Runtime adaptation	FC, SMM, MPM, ARE, BBS, RAM

The requirements identified in [30] can be classified into five categories: scalability, runtime adaptation, correctness, non-intrusiveness, and security. Table IX [1] analyzes how the proposed mechanisms support the requirements that are in addition to [29]: security and runtime adaptation.

Scalability is a key property in any distributed monitoring system. Real applications and systems are often quite dynamic and the scale of use can change rapidly. Scalability refers not only to scaling from a small system to a large one but also to scaling from large to small, and to scaling in geographic coverage [31]. Scalability goals can also concern the use of a large number of applications in the same system. All scalability goals imply the need for flexibility in terms of how the monitoring tool and 'measurer nodes,' and their communication, are constructed. GEMOM basic solutions, which comprise the MOM approach and the inherently encapsulated publish/subscribe communication mechanism, support scalability well [1]. More effort to ensure scalability is needed in other kinds of measurement architectures with complex probe management.

If necessary, the hierarchical architecture of Monitors and MTs can be created to support a large number of measurements. It is easy and straightforward to configure and initiate new Monitors. In principle, the chosen solutions do not set constraints on the scale of the system, in terms of the number of nodes or from a geographic coverage perspective. If high-volume measurements are needed, however, a special type of measurement channel solution is required.

The performance of the overall monitoring system is constrained by the underlying network solutions, typically the Internet networks in use. Network management is a problem for all networks as they grow in size. It may also be challenging to use a huge number of topics or namespaces, with proper authentication and authorization management. The number of measurement topics and namespaces can be kept under control with informed and adaptive planning and configuration. A large number of overlapping publish and/or subscribe actions result in a need for sophisticated mutual exclusion management solutions. Chockler et al. [32] suggested techniques for scalable solutions for publish/subscribe activity with many topics [1].

As GEMOM and its applications are run on various platforms that range from desktop computers to smart mobile phones, the portability of the monitoring solutions is important. The underlying MOM middleware solution enables applications to establish communication and interaction without having knowledge of the platform on which the application resides within the network. The monitoring approach is built using the same communication approach as the system in general. This means that the portability of the monitoring approach is good; only appropriate interfaces between the GEMOM nodes and their repositories and the services of the platform need to be implemented [1].

Interoperability of the GEMOM monitoring solution with other security tools is desirable in order to obtain a more holistic 'security picture' of the environment. The target of security monitoring can be applications, hosts, and the network. The GEMOM monitoring solution emphasizes those applications that operate in the GEMOM system, as well as host-based performance, resilience, and self-protection information [1].

Tools that focus on Internet-based traffic include vulnerability scanners, packet sniffers, various kinds of traffic analyzer tools, and application-specific scanners. A variety of these tools is available as commercial and open products. These tools can be connected to the MT in a straightforward manner using the publish/subscribe

mechanism. They often do not support interpretation of the data or correlation of input or logs from different sources however. In other words, they concentrate on the raw measurements. Almost all of the tools in the area of Internet Protocol (IP) traffic measurement and analysis perform only a small subset of the functionalities required to capture, file and classify, store, analyze traffic, and prepare the results for graphical display or for export into a database or other framework [33]. Consequently, the seamless integration of these tools to the GEMOM monitoring approach requires the development and configuration of specific Monitors that are capable of interpreting and correlating data produced by the tools [1].

The Open Web Application Security Project (OWASP) [17] and Insecure.org [34] list and analyze different security tools, some of which are potentially useful for complementing the GEMOM adaptive security management environment.

VI. Related Work

The US National Institute of Standards and Technology's (NIST) security metrics report [4] believes that the development of security-measurability-enhancing mechanisms is a promising research direction. Few research results are available in this new field however. In the following, we discuss related research that suggests mechanism for enhancing security measurability.

Martin [23] focused on the issues highlighted in Section III.H of this paper in his discussion on security-measurability-enhancing mechanisms based on SCAP enumerations and scoring systems. He elaborated on the mechanisms from an organizational perspective and showed how these mechanisms can be integrated into risk management, operations security management processes and enterprise networks. He does not deal with security-measurability-enhancing mechanisms from a technical perspective however.

Ciszkowski et al. [35] described an end-to-end quality and security monitoring approach for a Voice-over-Internet Protocol (VoIP) service over Peer-to-Peer (P2P) networks, providing adaptive QoS and DoS/Distributed DoS attack detection. They also introduced an associated trust and reputation framework to support routing decisions. The standalone modules in this architecture include Security, QoS, Reputation Management, and Intrusion Detection functionalities, the communication of which is arranged via channels. These choices do not allow as much flexibility and scalability as the GEMOM dynamical monitoring.

Jean et al. introduced a distributed and adaptive security monitoring system based on agent collaboration in [36], using a special algorithm to classify malicious agents based on their execution patterns. They also used the notion of an agent's security level but did not provide further details of the parameters of the security level calculation. They also defined a trust and confidence notion for the hosts of the agents.

Spanoudakis et al. [37] introduced a runtime security monitoring system based on confidentiality, integrity, and availability patterns. Their architecture contains a Monitoring Manager that takes requirements as an input and controls Monitoring Engine. Measurement Agents deliver the measured data to an Event Catcher, communicating with the Monitoring Engine. This architecture is interesting because it can be mapped directly to the more flexible GEMOM monitoring architecture. A special monitoring pattern language is used to define the pattern metrics for the Monitoring Manager. This work is limited to the basic abstract dimensions of security – confidentiality, integrity, and availability.

Kanstrén and Savola described five-layer reference architecture for a secure and adaptive distributed monitoring framework in [28], as developed in the BUGYO Beyond Research project. The main approach of this reference architecture is to increase non-intrusiveness by isolating the monitoring framework from the observed system. The architecture uses abstraction layers for the management of different practical measurement management objectives. In systems using measurement probes, this kind of architecture is usual, but for publish/subscribe-based systems like GEMOM, it can bring too much complexity.

Evesti et al. [38] proposed a preliminary environment for runtime security management that consists of service discovery, service adaptation, measurement, and decision-making functionalities. Vulnerability databases are used as a basis for measurements. The decision-making functionality can be seen as carrying out functions that are similar to those that the Security Measurement Manager and Overlay Manager perform in GEMOM.

Bejtlich [39] discussed network security monitoring at length and overviewed some tools and research solutions used, especially in traffic and packet monitoring. If the GEMOM environment is used over the Internet Protocol (IP), these tools can complement the GEMOM security monitoring environment by offering more details of IP traffic. Bejtlich claimed that traditional Intrusion Detection Systems (IDSs) do not deliver the value promised by vendors and that detection techniques that view the alert as the end goal are doomed to fail. Like GEMOM's monitoring and adaptive security management approaches, Bejtlich views alert data as an indicator and as the beginning of the actual decision-making process. Intrusion Detection and Prevention Systems (IDS/IPS) must also adapt to changes in the threat and vulnerability environment.

Bulut et al. present a measurement framework for security assurance in [40]. The framework collects information about the security-enforcing mechanisms of the system under investigation. The framework contains three different types of components: probe-agents, multiplexing agents and server-agents. Multiplexing agents are responsible for multiplexing data over subnet boundaries, and server-agents handle centralized processing of the measured data.

Vandelli et al. present a measurement framework for scientific experiments in [41] that are capable of capturing massive amounts of data. Standardized architectures and protocols are used between the measurement components. A separate data stream is used to pass high-volume data to the measurement system, and a separate channel is used to pass

control requests. This kind of high-performance measurability-enhancing solutions can be used in data gathering of security-relevant log information. In most security measurements, high-volume data do not need to be transferred.

A survey of approaches to adaptive application security and adaptive middleware can be found in [42] and [43], respectively.

VII. DISCUSSION

Security is a multi-faceted socio-technical challenge, and despite the long record of academic research and wide experience of practical issues, it still suffers from systematic and widely accepted design and management techniques. Security measurements introduce systematic thinking to security issues. However, nowadays, the use of security metrics and measurements is hard due to the lack of information and enough support from developed systems. It is clearly a 'chicken or egg' problem: in order to advance the field of security metrics, you need evidence from the actual system, and in order to be able to gather that evidence (and justify the gathering effort), you need metrics. Security-measurability-enhancing mechanisms aim to make evidence gathering effective and cost-effective, and they contribute especially to improved availability and attainability of the evidence. If these mechanisms become more widely accepted and applied, the field of security metrics can definitely make big steps forward. The fact that public/subscribe communication was chosen as the underlying communication paradigm is an important design choice supporting the measurability goals well. Flexibility in measurements is needed to 'pave the path' for wider acceptance of the use of security metrics and measurements.

We emphasize that the collection of security-measurability-enhancing mechanisms discussed in this study is preliminary. It is obvious that specific security measurement needs will raise the need for new and different kinds of mechanisms. In the future, it makes sense to also carry out standardization efforts in this field; it is easier to support the wide acceptance of the mechanisms if there are commonly agreed ways to do so. Recent advances in information security management system (ISO/IEC 27000 series) standardization have shown that there is interest in incorporating security metrics and measurements in security management standards. The current work in a standardization world regarding security metrics is quite vague. Nowadays, there are still big gaps between security management and engineering-oriented standards, the former concentrating on a top-down approach, and the latter mainly on bottom-up check-lists. Metrics, with the help of enough support of security-measurability-enhancing mechanisms, can play the role of bridging this gap. Wide acceptance of security metrics and measurement approaches can make remarkable advances in the whole security field.

VIII. CONCLUSIONS AND FUTURE WORK

We have discussed solutions to enhance the security measurability of telecommunications and software-intensive systems. The presented solutions have most value if they are built into the system under investigation already during the architectural design of it. The solutions were discussed in the context of the distributed messaging system GEMOM incorporating adaptive security management functionality. Solutions proposed and discussed in this study:

- *A flexible communication mechanism* is crucial to the security measurements. In GEMOM, the use of the publish/subscribe mechanism for measurement reporting increases the flexibility, scalability, and effectiveness of security measurements. In other types of communication architectures, measurement probes should be designed hand-in-hand with the architectural choices of the system.
- *Data and functionality redundancy* can help in obtaining valuable security-relevant data from fault situations. Redundancy can be implemented by smart maintenance of Mirror Brokers and mirrored measurement data, as independently as possible from the main Brokers and measurement data.
- *Multi-point measurement support* is often needed in security measurements, as only holistic security evidence is meaningful and the functionalities and processes of various system components and stakeholders contribute to that evidence. Holistic security measurability can be implemented by deploying several monitoring tools that are able to communicate continuously with each other.
- *Auto-recovery on error mode* helps in the investigation of failure before resuming normal operation. It is a helpful functionality for deeper analysis of attacks, such as root-cause analysis.
- *Configuration correctness, integrity and availability checks* of software constructs and input, buffers, memory, storage, and databases are integral parts of the security measurability solution.
- *State of security framework* supports categorization of the metrics and measurements with respect to time. Timing of different measurements and metrics is also important; it is necessary to define the state of security in order to establish proper time-dependency for the measured information.
- *Seamless co-operation of the security building blocks* at system architectural and process level should include a systematic and automatic information exchange. In order to ensure enough security measurability support, these issues should be taken into account already during the architectural design phase of the system and during the risk management and security assurance activities.
- The use of *shared metrics and measurement repositories and enumerations* supports the development of security metrics and ensures that the monitoring system has up-to-date threat and vulnerability knowledge. Security effectiveness metrics, in particular, can be based on repositories and enumerations.
- *The use of measures developed for other measurements* can be useful for security measurements. Some parameters that were originally developed for other measurement purposes can be applied to security

measurements. These include QoS, performance indicators, delay, delay variation, and packet loss rate, along with other indicators that reflect abnormal operation. Adequate authentication and authorization mechanisms ensure that the necessary data are available from modules for which secure coding principles are thoroughly enforced.

Our future work includes using the monitoring system for adaptive security management in experimental GEMOM system application use case investigations in critical information systems. This experimentation work will analyze the feasibility of the monitoring approach, the security metrics that are used, and the identified security-measurability-enhancing mechanisms.

ACKNOWLEDGMENTS

The main part of the work presented in this study was carried out in the GEMOM FP7 research project (2008–2010), which was partly funded by the European Commission. Some of the analytical work in study was carried out in the BUGYO Beyond CELTIC Eureka project (2008–2011). The authors acknowledge the contributions to the GEMOM system description made by various GEMOM partners, and collaboration with persons working in the BUGYO Beyond project.

REFERENCES

[1] R. Savola and P. Heinonen, "Security-Measurability-Enhancing Mechanisms for a Distributed Adaptive Security Monitoring System," SECURWARE 2010, Venice/Mestre, Italy, July 18-25, 2010, pp. 25-34.

[2] H. Abie, I. Dattani, M. Novkovic, J. Bigham, S. Topham, and R. Savola, "GEMOM – Significant and Measurable Progess Beyond the State of the Art," ICSNC 2008, Sliema, Malta, Oct. 26–31, 2008, pp. 191–196.

[3] H. Abie, R. Savola, and I. Dattani, "Robust, Secure, Self-Adaptive and Resilient Messaging Middleware for Business Critical Systems," ADAPTIVE 2009, Athens/Glyfada, Greece, Nov. 15–20, 2009, pp. 153–160.

[4] W. Jansen, "Directions in Security Metrics Research," U.S. National Institute of Standards and Technology (NIST), NISTIR 7564, Apr. 2009, 21 p.

[5] R. Savola and H. Abie, "Development of Security Metrics for a Distributed Messaging System," AICT 2009, Baku, Azerbaijan, Oct. 14–16, 2009, 6 p.

[6] R. Savola, "A Security Metrics Taxonomization Model for Software-Intensive Systems," Journal of Information Processing Systems, Vol. 5, No. 4, Dec. 2009, pp. 197–206.

[7] R. Savola and H. Abie, "Development of Measurable Security for a Distributed Messaging System," International Journal on Advances in Security, Vol. 2, No. 4, 2009, pp. 358–380 (March 2010).

[8] G. Jelen, "SSE-CMM Security Metrics," NIST and CSSPAB Workshop, Washington, D.C., USA, June 2000.

[9] R. Savola and H. Abie, "Identification of Basic Measurable Security Components for a Distributed Messaging System," SECURWARE '09, Athens/Glyfada, Greece, Jun. 18–23, 2009, pp. 121–128.

[10] D. S. Herrmann, "Complete Guide to Security and Privacy Metrics – Measuring Regulatory Compliance, Operational Resilience and ROI," Auerbach Publications, 2007, 824 p.

[11] A. Jaquith, "Security Metrics: Replacing Fear, Uncertainty and Doubt," Addison-Wesley, 2007.

[12] N. Bartol, B. Bates, K. M. Goertzel, and T. Winograd, "Measuring Cyber Security and Information Assurance: a State-of-the-Art Report," Inf. Assurance Tech. Analysis Center IATAC, May 2009.

[13] C. Wang and W. A. Wulf, "Towards a Framework for Security Measurement," 20th National Information Systems Security Conference, Baltimore, MD, Oct. 1997, pp. 522–533.

[14] J. R. Williams and G. F. Jelen, "A Framework for Reasoning about Assurance," Arca Systems, Inc., 1998, 43 p.

[15] T. Kanstrén et al., "Towards an Abstraction Layer for Security Assurance Measurements (Invited Paper)," MeSSA '10, Proc. 4th European Conf. on Software Architecture: Companion Volume, pp.189-196.

[16] G. H. Kim and E. H. Spafford, "The Design and Implementation of Tripwire: a System Integrity Checker," Computer and Communications Security, Fairfax, VA, Nov. 2–4, 1994, pp. 18-29.

[17] OWASP, Open Web Application Security Project. [On-line]. Available: www.owasp.org [Accessed June 20, 2011].

[18] E. A. Nichols and G. Peterson, "A Metrics Framework to Drive Application Security Improvement," IEEE Security & Privacy, Mar./Apr. 2007, pp. 88–91.

[19] M. Howard and D. LeBlanc, "Writing Secure Code," Microsoft Press, 2003, 768 p.

[20] M. Schiffman, G. Eschelbeck, D. Ahmad, A. Wright, and S. Romanosky, "CVSS: A Common Vulnerability Scoring System," U.S. National Infrastructure Advisory Council (NIAC), 2004.

[21] R. A. Martin, "Managing Vulnerabilities in Networked Systems," IEEE Computer Society Computer Magazine, Vol. 34, No. 11, Nov. 2001.

[22] M. Barrett, C. Johnson, P. Mell, S. Quinn, and K. Scarfone, "Guide to Adopting and Using the Security Content Automation Protocol (SCAP)," NIST Special Publication 800-177 (Draft), U.S. National Institute of Standards and Technology, 2009.

[23] R. A. Martin, "Making Security Measurable and Manageable," MILCOM '08, Nov. 16–19, 2008, pp. 1–9.

[24] Web Application Security Consortium (WASC), "Threat Classification," Version 2.0. [Online]. Available: www.webappsec.org [Accessed June 20, 2011].

[25] SANS Institute, "The Top Cyber Security Security Risks." [Online]. Available: www.sans.org/top-cyber-security-risks/ [Accessed June 20, 2011].

[26] R. A. Martin, "Making Security Measurable and Manageable," MILCOM '08, San Diego, California, Nov. 16-19, 2008, 9 p.

[27] R. Savola and T. Frantti, "Core Security Parameters for VoIP in Ad Hoc Networks," WPMC '09, Sendai, Japan, Sep. 7–10, 2009, 5 p.

[28] R. Seacord, "CERT Top 10 Secure Coding Practices." [Online]. Available: www.securecoding.cert.org [Accessed June 20, 2011].

[29] R. Savola, "On the Feasibility of Utilizing Security Metrics in Software-Intensive Systems," International Journal of Computer Science and Network Security, Vol. 10, No. 1, Jan. 2010, pp. 230–239.

[30] T. Kanstrén and R. Savola, "Definition of Core Requirements and a Reference Architecture for a Dependable, Secure and Adaptive Distributed Monitoring Framework," DEPEND 2010, Venice, Italy, July 18–25, 2010, 10 p.

[31] P. Venables, "Security Monitoring in Heterogeneous Globally Distributed Environments," Information Security Technical Report, Vol. 3, No. 4, 1998, pp. 15-31.

[32] G. Chockler, R. Melamed, Y. Tock, and R. Vitenberg, "Constructing Scalable Overlays for Pub-Sub with Many Topics – Problems, Algorithms and Evaluation," PODC '07, Portland, Oregon, USA, 2007, pp. 109–118.

[33] J. Quittek, A. Bulanza, S. Zander, C. Schmoll, M. Kundt, E. Boschi, and J. Sliwinski, "MOME Project – State of Interoperability," MOME Project WP1 Deliverable 11, Technical Report, 2006.

[34] Insecure.org [On-line]. Available: www.insecure.org [Accessed June 20, 2011].

[35] T. Ciszkowski, C. Eliasson, M. Fiedler, Z. Kotulski, R. Lupu, and W. Mazurczyk, "SecMon: End-to-End Quality and Security Monitoring System," 7th Int. Conf. on Computer Science, Research and Applications (IBIZA '08), Kazimierz Dolny, Poland, Jan. 31–Feb. 2, 2008. Published in Annales UMCS, Informatica, AI 8, pp. 186–201.

[36] E. Jean, Y. Jiao, A. R. Hurson, and T. E. Potok, "Boosting-based Distributed and Adaptive Security-Monitoring through Agent Collaboration," IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology 2007 – Workshops, pp. 516–520.

[37] G. Spanoudakis, C. Kloukinas, and K. Androutsopoulos, "Towards Security Monitoring Patterns," ACM Sympos. on Applied Computing, Seoul, Korea, 2007, pp. 1518–1525.

[38] A. Evesti, E. Ovaska, and R. Savola, "From Security Modelling to Run-time Security Monitoring," SEC-MDA, Enschede, Netherlands, Jun. 24, 2009, pp. 33–41.

[39] R. Bejtlich, "The Tao of Network Security Monitoring – Beyond Intrusion Detection," Addison-Wesley, 2004, 798 p.

[40] E. Bulut, D. Khadraoui, and B. Marquet, "Multi-Agent Based Security Assurance Monitoring System for Telecommunication Infrastructures," Proc. Communication, Network and Information Security, 2007.

[41] W. Vandelli et al., "Strategies and Tools for ATLAS Online Monitoring," IEEE Transactions on Nuclear Science, Vol. 54, No. 3, pp. 609-617, June 2007.

[42] A. Elkhodary and J. Whittle, "A Survey of Approaches to Adaptive Application Security," SEAMS '07, May 20–25, 2007.

[43] S. Sadjadi, "A Survey of Adaptive Middleware," Technical Report MSU-CSE-03-35, Michigan State University, East Lansing, Michigan, Dec. 2003.

Estimating Human Movement Parameters Using a Software Radio-based Radar

Bruhtesfa Godana*, André Barroso†, Geert Leus‡

* Norwegian University of Science and Technology, Trondheim, Norway
†Philips Research Europe, Eindhoven, The Netherlands
‡Faculty of EEMCS, Delft University of Technology, Delft, The Netherlands
*godana@iet.ntnu.no, †andre.barroso@philips.com, ‡g.j.t.leus@tudelft.nl

***Abstract*—Radar is an attractive technology for long term monitoring of human movement as it operates remotely, can be placed behind walls and is able to monitor a large area depending on its operating parameters. A radar signal reflected off a moving person carries rich information on his or her activity pattern in the form of a set of Doppler frequency signatures produced by the specific combination of limbs and torso movements. To enable classification and efficient storage and transmission of movement data, unique parameters have to be extracted from the Doppler signatures. Two of the most important human movement parameters for activity identification and classification are the velocity profile and the fundamental cadence frequency of the movement pattern. However, the complicated pattern of limbs and torso movement worsened by multipath propagation in indoor environment poses a challenge for the extraction of these human movement parameters. In this paper, three new approaches for the estimation of human walking velocity profile in indoor environment are proposed and discussed. The first two methods are based on spectrogram estimates whereas the third method is based on phase difference computation. In addition, a method to estimate the fundamental cadence frequency of the gait is suggested and discussed. The accuracy of the methods are evaluated and compared in an indoor experiment using a flexible and low-cost software defined radar platform. The results obtained indicate that the velocity estimation methods are able to estimate the velocity profile of the person's translational motion with an error of less than 10%. The results also showed that the fundamental cadence is estimated with an error of 7%.**

***Index Terms*—Human motion, Human gait, Velocity profile, Cadence frequency, Radar, GNU Radio**

I. Introduction

Automatic classification of human activity is an enabler of relevant applications in the healthcare and wellness domains given the strong empirical relation between a person's health and his or her activity profile. As a rule of thumb, the ability of a person to engage independently in strenuous and complex activities entails better fitness and health status, the reverse relation being also generally true. This implication has inspired the design of activity monitoring systems that range from fitness training [3] to early discharge support of postoperative patients [4]. Seniors living independently by wish or circumstances may also benefit from remote activity classification as a means of assessing their health status or identifying accidents and unusual behaviour [5]. This information can be fed to companies specialized in providing swift help in case of need, healthcare providers or concerned family members.

On-body or off-body sensors can be used for human activity monitoring in indoor environment. In the former category, triaxial accelerometers have been widely investigated for quantifying and classifying human activities [6]. The main drawback of on-body sensors is that these must be carried by the monitored subject at all times. In elderly care applications, where long monitoring periods are expected, subjects can be forgetful or uncooperative thus hampering the monitoring process. In the latter category, off-body sensing for movement analysis can be performed using technologies such as cameras [7], ultrasound [8] or pyroelectric infrared (PIR) sensors [9]. These approaches suffer however from limited range indoors as line of sight is usually constrained to a single room. The range limitation of these technologies means that many sensors are required to cover a single building. Furthermore, these multiple sensing units must be networked for data collection thus increasing the deployment and maintenance complexity of the system. Radar on the other hand is an attractive technology for long term monitoring of human movement because it does not need to be carried by the user, can be placed behind walls and is able to cover a large area depending on its operating parameters. Furthermore, the coarseness of the information provided by radars is less prone to raise privacy concerns when compared to cameras. Depending on the operating parameters, radars can also be used for through-the-wall sensing [10].

Deploying radars in health and wellness applications at the user's home will be facilitated if such systems are low cost, easy to deploy and safe. The possibility to adapt simple wireless LAN transceivers into indoor radars keeps the radar cost low and makes it flexible. Low radiation emission ensures safety for the user while multiple room coverage per radar unit eases deployment at home. However, extracting useful information from radars deployed in an indoor environment, where subjects may spend most or all their time, poses a challenge due to multipath propagation, presence of walls and other big objects, presence of interfering motions, etc. These properties of an indoor environment make it difficult to identify patterns of human movement from an indoor radar signal. Though these issues are addressed in this paper, the presence of interfering motions is not considered. In this work, a low-cost radar is designed that extracts human movement parameters in the presence of indoor multipath and clutter.

A radar signal reflected of a moving person carries rich information on his or her activity in the form a set of Doppler frequency patterns produced by the specific combination of limbs and torso movements. The Doppler frequency pattern that results from such a complex movement sequence is called "micro-Doppler signature" and the movement pattern is called "gait". If for a given activity, these Doppler signatures can be categorized into unambiguous profiles or "footprints", then radar signals can be used to identify the occurrence of specific activities over time. The evolution of these micro-Doppler patterns over time can be viewed in a spectrogram which is a time versus Doppler frequency plot of the micro-Doppler signatures. Spectrogram patterns obtained from human movement contain rich information on different parameters of movement including direction of motion, velocity, acceleration, displacement, cadence frequency, etc. Therefore, a visual inspection of spectrogram patterns reveals the occurrence of different types of human activities. However, to enable automatic human activity classification, parameters that have a unique range for the different types of human activities must be extracted from the micro-Doppler signature. Moreover, data storage and transmission of an entire spectrogram plot consumes too much storage and transmission resources. For efficient storage and transmission of human movement data to care taking centres, unique parameters that enable classification and require less transmission resources should be selected.

One of the most important parameters for the classification of human activities using Doppler signatures is the velocity profile [11], *i.e.*, the instantaneous velocity of human motion over time. Moreover, the velocity profile of a walking person shows different states (accelerate, decelerate, sudden stop, change in direction, etc.) that are useful to be identified in various applications. In general, a careful observation of how a person's velocity profile develops over time provides insights that can be used for timely intervention (if and when needed) in health and elderly care applications. Another important parameter for human activity classification is the rate of oscillation of the limbs which is called the "fundamental cadence frequency". This is an average rather than instantaneous parameter which shows how fast the legs and arms of a person are oscillating. The fundamental cadence frequency is an important parameter which can be directly utilized by an activity classification system [12], [11].

In this paper, different approaches to estimate these two important parameters of human motion, namely velocity profile and fundamental cadence frequency, are proposed and evaluated. The main contributions of this paper to the area of unobtrusive monitoring in health and wellness applications are as follows:

- Two different methods to estimate the velocity profile of human translational motion from the Doppler signature obtained in a form of time-frequency spectrogram are proposed and evaluated. The possibility of using high resolution Doppler spectrum estimation techniques is also introduced.
- A third simple method to estimate the velocity profile of human motion based on phase difference computation is suggested and evaluated.
- An experimental radar platform based on low-cost software-defined radio hardware and open source software is implemented and its use for indoor monitoring of human movement is validated. The platform offers the opportunity of realizing low-cost experiments at an expedited pace and low budget.

The remainder of this paper is organized as follows: Section II reviews related work in the area of using radars for human activity monitoring, characterization and classification. Section III describes a human movement model that is crucial for the identification of the major Doppler components in the radar signal. Section IV introduces basic radar concepts in human sensing such as human radar cross-section and the radar signal model. Section V discusses the pre-processing and spectral estimation techniques that are relevant to obtain the micro-Doppler signatures. The proposed velocity profile and cadence frequency estimation methods are discussed in Sections VI and VII respectively. Section VIII describes the software defined radar platform and the experimental setup used in the validation experiments. The estimation results are presented and evaluated in Section IX. Finally, Section X summarizes and concludes the paper.

II. Related Work

Human detection using radars has been extensively researched for military surveillance and rescue applications [13][14][10][15]. The use of radars for human activity monitoring and classification has also been intensively investigated. Anderson [16] used multiple frequency continuous wave radar for classification of humans, animals and vehicles. Otero [12] used a $10\,GHz$ CW radar using micro-path antennas to collect data and to attempt classification. In addition [12] introduced a technique to estimate the cadence frequency of motion. Gurbuz et al. proposed a simulation based gender discrimination using spectrogram of radar signals [17]. Hornsteiner et al. applied radars to identify human motion [18]. Kim et al. used artificial neural network for classifying human activities based on micro-Doppler signatures [11]. All these papers used Fast Fourier Transform based frequency estimation.

There is also previous work on using other transforms for Doppler pattern estimation. Geisheimer et al. [19] introduced the chirplet transform as spectral analysis tool. The Hilbert-Haung Transform for non-linear and non-stationary signals in wide band noise radars is also suggested by Lay et al. [20]. A complex but more accurate iterative way to obtain each pixel in the spectrogram in a bid to improve the frequency resolution and suppress the side lobes of the Fast Fourier Transform is also suggested by Du et al. [21].

Even though the above authors have treated different aspects in human activity classification in general, the estimation of velocity profile in indoor environment where the received signal is plagued with multipath propagation was not specifically treated. Recently, spectrogram based methods to estimate the velocity profile of human walking were proposed in [1]. A

displacement estimation method based on computing phase difference is also proposed in [2].

In this paper, the spectrogram estimation methods in [1] are compared with another velocity profile estimation method derived from the the phase difference principle in [2]. The use of sliding window high resolution parametric spectral estimator (MUSIC) is introduced and its performance for velocity profile estimation is compared with the commonly used Fast Fourier Transform. Moreover, a cadence frequency spectrogram is estimated and a simple method to estimate the fundamental cadence frequency from the spectrogram is suggested and evaluated.

III. Human Movement Model

Our starting point for human activity characterization is the definition of a movement model. After studying the relationship between the different parts of the body during locomotion, features that have unique values in different activities can be identified. In this regard, the person's velocity profile is one of the important features that can be used to achieve activity classification.

The velocity profile refers to the instantaneous temporal displacement that the different parts of the human body attain during movement. Most of the human movement models available rely on dividing the non-rigid human body into the most significant rigid body parts and modelling the velocity profile of these rigid components. One of the most used human movement models [22] decomposes the body into 12 parts consisting of the torso, lower and upper part of each leg, lower and upper part of each arm, the head and each of the right and left foot. The torso is the main component or trunk of the body. This model also describes the kinematics of each of these body parts as a person walks with a particular velocity. Another known model was based on 3-D position analysis of reflective markers worn on the body using high resolution camera [23]. This model states that the velocity profile of each body part can be represented using low-order Fourier series. Using this model as a basis, we have described a modified human movement velocity profile as follows.

Assume a person is moving at a constant velocity V in a certain direction and that the human body consists of M rigid parts. The velocity profile of each part, $V_m(t)$, can be represented as a sum of sinusoids given by:

$$\begin{aligned} V_m(t) &= V + A\{k_{m1}\sin(\omega_c t + p_m) + \\ &\quad k_{m2}\cos(\omega_c t + p_m) + k_{m3}\sin(2\omega_c t + p_m) + \\ &\quad k_{m4}\cos(2\omega_c t + p_m)\} \end{aligned} \quad (1)$$

where $1 \leq m \leq M$. Note that the velocity profile of each body part V_m is characterized by amplitude constants: $k_{m1}, ..., k_{m4}$ and a phase constant: p_m $(0 \leq p_m \leq 180^o)$. The oscillation amplitudes $k_{m1}, ..., k_{m4}$ are largest for legs and smallest for the torso. The phase p_m reflects the locomotion mechanism of the body. For instance, the right leg and left arm combination move 180^o out of phase with respect to the left leg and right arm. A is a constant that has a specific value for different types of human activities, ω_c is the frequency of oscillation of the body parts which is also called the fundamental cadence frequency of motion.

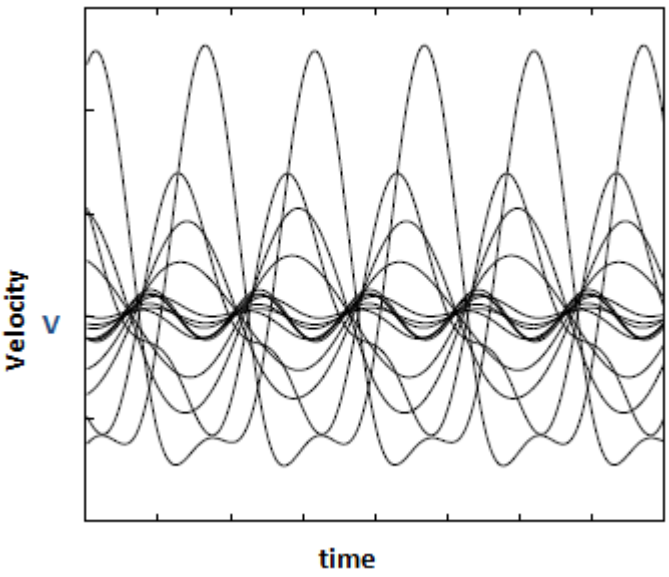

Figure 1. Human walking velocity profile model [18]

A simulation of the velocity profile of a walking person based on a model similar to the one stated above is shown in Figure 1. As the Figure shows, the amplitude of oscillation of each body part is different; however, all the body parts oscillate at the same fundamental frequency ω_c and its second harmonics $2\omega_c$.

The translational velocity of the body is normally time-varying. Therefore, the oscillations of the body parts in (1) will be superimposed on the time varying velocity profile of the body. The torso has the smallest oscillation amplitudes, $k_{m1}, ..., k_{m4}$ and therefore the translational velocity profile V of the body can be approximated by the velocity of the torso. The translational velocity can thus be obtained by estimating the velocity of the torso. Therefore, the two terms: velocity profile of the body and velocity profile of the torso are assumed to be the same and used interchangeably from now on.

The velocity profile of the other parts of the body, $V_m(t)$ can thus be expressed as sinusoids superimposed on the velocity profile of the torso. Therefore, (1) can be expressed as:

$$\begin{aligned} V_m(t) &= V_{torso}(t) + A\{k_{m1}\sin(\omega_c t + p_m) + \\ &\quad k_{m2}\cos(\omega_c t + p_m) + k_{m3}\sin(2\omega_c t + p_m) + \\ &\quad k_{m4}\cos(2\omega_c t + p_m)\} \end{aligned} \quad (2)$$

IV. Radar in human sensing

Radar is a device that transmits electromagnetic waves, receives the signal reflected back off the target and extracts information about the characteristics (range, velocity, shape, reflectivity, etc.) of the target. The amount of electromagnetic

energy that a target is capable of reflecting back is measured in terms of the radar cross section of the target. Doppler radars are those that measure the velocity of a target based on the Doppler effect, *i.e.*, an electromagnetic wave hitting a moving target undergoes a frequency shift proportional to the velocity of the target. The radar cross section and velocity profile are constant and easy to determine for a rigid body moving at a constant speed. However, as discussed in Section III, the human body locomotion is more complicated. The radar cross section and the signal model for radar based human movement monitoring are discussed in the following sections.

A. Human Body Radar Cross Section

Radar cross section (RCS) is a measure of signal reflectivity of an object and is usually expressed in a unit of area (e.g., m^2). RCS depends on the frequency of the transmitted signal and parameters of the target such as size, shape and material [24]. The RCS of a moving person is challenging to model because the human body is composed of multiple semi-independent moving parts. A simple additive approach to create an RCS model by adding up the contribution of each body part is commonly adopted. The contribution of each part can be assumed to remain constant during motion without significant error. In addition, the total RCS can be assumed to be half of the body surface area which is exposed when the person is facing the radar; this area is typically listed as $1m^2$ [25]. Each of the 12 major parts of the human body listed in Section III contribute to a fraction of the RCS. The torso has the highest RCS followed by the legs and arms. The head and feet have the least contribution. Particularly, the percentage contribution of each body part is listed as: torso 31%, arms 10% each, legs 16.5% each, head 9% and feet 7% [25].

As the torso has the highest RCS of all the moving body parts, the velocity profile of the torso can in principle be estimated by picking out the strongest component from the received Doppler signal.

B. Signal Model

Doppler radars measure the frequency shift of electromagnetic waves due to motion. The Doppler shift of an object is directly proportional to the velocity of the object and the carrier frequency of the transmitted signal as described below.

Assume a narrowband, unmodulated signal $a\, e^{j(2\pi f t+\phi_0)}$ is transmitted where a, f and ϕ_0 are the amplitude, carrier frequency and initial phase respectively. The signal received at the receiver antenna being reflected off a person has a time varying amplitude $a(t)$ and a time varying phase $\phi(t)$; thus it is given by $a(t)\, e^{j(2\pi f t+\phi_0+\phi(t))}$. Hence, the received baseband signal after demodulation reduces to:

$$y(t) = a(t)\, e^{j\phi(t)} \tag{3}$$

The Doppler frequency shift, f_d is the rate of change of the phase of the signal, *i.e.*, $f_d(t) = -\frac{1}{2\pi}\cdot\frac{d\phi(t)}{dt}$ and a small change in phase can be expressed in terms of the change in distance as $\frac{d\phi(t)}{dt} = \frac{4\pi}{\lambda}\frac{dR(t)}{dt}$ where $R(t)$ represents the distance. This implies that the Doppler shift of a rigid target moving at a velocity $V(t)$ is given by $f_d(t) = 2\frac{V(t)}{\lambda}$ where λ is the wavelength of the transmitted radio wave and the velocity $V(t)$ represents $-\frac{dR(t)}{dt}$. This is in a mono-static radar configuration where the transmitter and receiver are co-located. In bi-static configuration where the transmitter and receiver are located on opposite sides of the target, the Doppler shift is given by $f_d(t) = \frac{V(t)}{\lambda}$.

It is stated in Section III that the different rigid components of the body have their own time-varying velocity profile superimposed on the body velocity. Therefore, each of these body parts have their own time-varying Doppler shift, *i.e.*, $f_{d_m}(t) = 2\frac{V_m(t)}{\lambda}$ where $V_m(t)$ is the velocity profile of each body part. It is however generally challenging to extract the velocity profiles of each body part for the following reasons:

- The received signal is a superposition of signals that consist of Doppler shifts of different moving parts. Moreover, each body part has different RCS resulting in different contribution to the aggregate signal.
- There is significant multipath fading in indoor environment which results in further additive components to the resulting signal.
- A radar measures only the radial component of the velocity of the person, and thus only a portion of the movement can be estimated with signals from a single radar.

The content that follows emphasizes on how to estimate the velocity profile of the body from the aggregate received signal.

A typical walking of a person in an indoor environment is described by non-uniform motion, *i.e.*, the velocity profile of the body varies with time. However, physical constraints limit the person from changing velocity during very short time intervals. Consequently, the person's velocity can be assumed to remain constant during short time intervals. In other words, a non-uniform human motion can be viewed as a uniform motion over small time or displacement intervals. This corresponds to the idea that the non-stationary radar signal received as a reflection from the person can be assumed to be piece-wise stationary. Based on this argument, the received signal during a small piece-wise stationary interval can be assumed to be a summation of a certain number of sinusoids. If D sinusoids are assumed, the received signal after sampling can be given by:

$$y\,[n] \;=\; \sum_{d=1}^{D}\left[a_d\cdot e^{j(\frac{4\pi V_d[n]T}{\lambda}n+\phi_d)}\right] \tag{4}$$

where $y\,[n]$ is a sample at time instant nT, T is the sampling time and a_d, V_d and ϕ_d are respectively the amplitude (which is proportional to the RCS), velocity and initial phase of each Doppler frequency component. Since the amplitude undergoes large scale variations as compared to the phase which varies from sample to sample, here it is assumed that the amplitude a_d is not time varying in the piece-wise stationary interval.

The indoor environment consists of stationary objects such as walls that have larger RCS than the human body. The

signal reflected from these stationary objects has zero Doppler frequency shift. Moreover, there is a strong direct signal between the transmitter and receiver antennas of the radar. The resulting effect is a strong DC component in the baseband radar signal. Therefore, the received radar signal is actually given by:

$$y[n] = a \cdot e^{j\phi} + \sum_{d=1}^{D} \left[a_d \cdot e^{j(\frac{4\pi V_d[n]T}{\lambda} n + \phi_d)} \right] \quad (5)$$

The number of sinusoids D may change between consecutive intervals, but it is assumed to remain constant to avoid complexity. The value of D can be taken as small as the number of body parts described in Section III; however, it is generally better to assign it a larger number to obtain a smooth Doppler spectrum pattern.

V. Doppler Spectrum Estimation

The received radar signal consists of many frequency components as described in the previous section. If piecewise stationarity is assumed, a joint time-frequency estimation can be used to decompose the received signal into these frequency components. In order to estimate the spectral content of a signal, non-parametric or parametric spectral estimators can be applied [26]. In this work, the Short Time Fourier Transform (STFT) and a high resolution parametric estimator, sliding window MUltiple SIgnal Classification (MUSIC) are used. However, as discussed in Section IV-B, the zero-frequency component which results due to stationary objects in the environment must be removed before spectrum estimation.

A. Pre-processing

As shown in (5), there is a strong DC component in the aggregate received signal. This component contains no information and makes the spectral magnitudes of the other relevant frequencies almost invisible in the spectrogram. Moreover, it affects estimation of the relevant Doppler frequency patterns which have small amplitudes. Therefore, this component must be removed for better estimation.

There are different techniques to eliminate a DC component from a signal. The simplest method available is adopted here, *i.e.*, averaging. The average value of the signal is computed and subtracted from the aggregate signal as follows:

$$\hat{y}[n] = y[n] - \frac{1}{N_{av}} \sum_{n=1}^{N_{av}} y[n] \quad (6)$$

where N_{av} is a large number. The remaining signal $\hat{y}[n]$ can be thus assumed to consist of the useful Doppler frequency pattern from moving objects only.

B. Spectrum Estimation

The short time Fourier transform (STFT) applied on the signal, $\hat{y}[n]$ is given by:

$$Y[k, n'] = \sum_{n=n'}^{n'+L} \hat{y}[n] \cdot e^{-j2\pi nk/N} \quad (7)$$

where L is the number of signal samples taken in each consecutive computation which is called "window size" in spectral estimation; n', which is set to multiples of $(1-\alpha)L$, represents the starting points of the moving window transform and α is the overlap factor between windows. k represents the k^{th} frequency component of the signal, and N is the size of the FFT. The window size L is set based on the duration over which the signal is assumed stationary. This form of short time FFT computation is also called sliding window FFT.

For the sake of comparison, a MUSIC [26] based spectral estimation is also applied to the received signal. MUSIC is a parametric spectral estimator based on eigenvalue decomposition. Sliding window MUSIC based spectral estimation is not commonly used; however, it is intuitive that it can be applied similar to the sliding window FFT. In the STFT, the window size is a trade-off between stationarity and spectral resolution. The major advantage of parametric spectral estimators like MUSIC is that the spectral resolution is independent of the window size L. However, the MUSIC method requires a priori knowledge of two parameters: the auto-correlation lag parameter and the number of sinusoids D [26]. The performance of the MUSIC method can be better or worse than STFT based on the setting of these two parameters.

The joint time-frequency spectral estimation is represented using the spectrogram, a color plot of the magnitude of frequency components as a function of time and frequency. The pixels in the spectrogram represent the power at a particular frequency and time, which is computed as: $P[k, n'] = |Y[k, n']|^2$.

VI. Velocity Profile Estimation Methods

As discussed in Section III, each body part has its own velocity profile superimposed on the velocity profile of the torso. The instantaneous torso velocity $v_{torso}[n']$ can be obtained from the instantaneous torso Doppler frequency $f_{torso}[n']$ using:

$$v_{torso}[n'] = \frac{\lambda}{2} f_{torso}[n'] \quad (8)$$

Three methods to estimate the velocity profile of human walking are suggested. The first two methods are based on the the joint time-frequency estimation discussed in Section V. The torso Doppler frequency profile is estimated using these two methods and the corresponding velocity profile is obtained using (8). The third method is different from the two methods. It is a simple but approximation-based method based on phase difference computation.

A. Maximum Power Method

As described in Section III, the torso has the largest RCS of all the body parts. Thus, the frequency component which has the highest power must be the Doppler frequency component of the torso since the strongest DC component is already removed. The maximum power method selects the frequency of maximum power from each spectral window in the computed

spectrogram, *i.e.*, $f_{torso}[n'] = f[k_{torso}, n']$, where k_{torso} is the frequency index at which $P[k, n']$ is maximum.

However, selecting the maximum frequency component returns the torso frequency component only when there is motion. If there is no motion, the received signal $\hat{y}[n]$ in (6) consists of only background noise and therefore selecting the strongest frequency component gives a wrong estimate of the torso frequency (which is actually zero). A threshold parameter must thus be selected to distinguish motion and no-motion intervals (for instance, in Figure 4, the interval of no-motion is 0-3 s). This parameter will be computed from the signal received when there is no motion and used as a threshold. The total signal power in the spectrogram column is one of the suitable parameters that can be used to distinguish these intervals. The parameter is computed and averaged over the duration of no-motion to determine a threshold, i.e., $P_{thr} = average_{n'}\{\sum_{k=1}^{N} P[k, n']\}$. Therefore,

$$f_{torso}[n'] = \begin{cases} f[k_{torso}, n'] & \text{if} \quad \sum_{k=1}^{N} P[k, n'] > P_{thr} \\ 0 & \text{else} \end{cases} \quad (9)$$

B. Weighted Power Method

The maximum power method requires a threshold which may fail to distinguish the motion and no-motion intervals correctly. This can result in a non-zero velocity estimate in absence of motion or zero velocity even though there is motion. Thus a method that pulls the velocity to zero when there is no or little motion without using a threshold is desirable. This method should also pull the resulting velocity estimate to torso velocity when there is motion.

One possible way to do this is to estimate $f_{torso}[n']$ as a power-weighted average frequency in each spectrogram column, n', *i.e.*,

$$f_{torso}[n'] = \frac{\sum_{k=1}^{N} f[k, n'] \cdot P[k, n']}{\sum_{k=1}^{N} P[k, n']} \quad (10)$$

This is based on the assumption that the frequency index range considered in the spectrogram is $[-Fs/2 : Fs/2]$ (where $Fs = \frac{1}{T}$ is the sampling frequency) or the zero frequency is the central point in the spectrogram.

The major problem of the weighted power method is that it results in a biased estimate when image frequencies are present. Image frequencies are those Doppler frequencies that occur on the opposite side of the actual Doppler frequency pattern in the spectrogram. These occur due to multipath effect in indoor environments. For instance, when a person is moving towards the radar, the Doppler frequencies are positive. However, there are also signals that reflect on the back of the person and received in the aggregate signal. As the person is moving away from the radar with respect to these signal paths, the signal components create negative (image) frequencies. The presence of image frequencies makes the weighted power estimate biased with respect to the actual torso frequency. However, the rays that reflect off the back of the person travel longer distances as compared to the rays that reflect off the front of the person and therefore, these components have lower power levels. The low power level of image frequencies reduces their impact on the weighted power.

The maximum power method is not affected by the presence of image frequencies as it simply selects the strongest frequency component. The weighted power method however performs well even in static conditions and is easier to apply as there is no need for a threshold.

C. Phase Difference Method

The third instantaneous velocity estimation method is derived from the total displacement estimation method suggested in [2] which was based on phase difference computation. In narrowband signals, the change in phase can be directly related to the propagation delay. Therefore, the change in phase can be directly related to the change in distance or the change in distance per unit time which is the instantaneous velocity.

After removing the DC component using (6), the received signal in (5) can be expressed as:

$$\hat{y}[n] = \sum_{d=1}^{D} \left[a_d \cdot e^{j(\frac{4\pi V_d[n]T}{\lambda} n + \phi_d)} \right] \quad (11)$$

Lets make a crude approximation that there is only one strong reflection in the received signal and all the other reflections are very weak. It is mentioned that if there is one strong component in the reflection from the human body, that strong component is the reflection from the torso. Using this assumption, (11) reduces to:

$$\hat{y}[n] \approx a_{torso} \cdot e^{j(4\pi V_{torso}[n]\frac{T}{\lambda} n + \phi_d)} \quad (12)$$

The instantaneous torso velocity can be easily be obtained from (12) by computing the phase difference between consecutive samples. The phase difference between consecutive samples $\Delta\phi[n]$ can be computed by:

$$\Delta\phi[n] = \angle(\hat{y}[n]\hat{y}^*[n-1]) \approx 4\pi V_{torso}[n]\frac{T}{\lambda} \quad (13)$$

This change in phase $\Delta\phi[n]$ should be very small here ($\Delta\phi[n] \ll 2\pi$) to avoid phase ambiguity. However, this is not a problem for typical sampling rates of a few hundred Hz and radar transmission frequencies less than 10 GHz which is also the case in our software radio-based radar.

Therefore, the torso velocity can be obtained as:

$$V_{torso}[n] \approx \Delta\phi[n]\frac{\lambda}{4\pi T} \quad (14)$$

It is discussed that human motion is piece-wise stationary; thus, a resolution more than a fraction of a second is not necessary. The motion is assumed to be stationary over L samples for spectrum estimation in Section V-B. Using a similar piece-wise stationarity range of L, the velocity profile of the torso is thus given by:

$$V_{torso}[n'] \approx \frac{\lambda}{4\pi LT} \sum_{n=n'}^{n'+L} \Delta\phi[n] \quad (15)$$

Besides estimating the velocity profile at an appropriate interval, the averaging in (15) has the advantage of averaging out the noise when there is no motion. Assuming that the noise is additive white noise when there is no motion (when the velocity is zero), the summation in (15) tends to zero. Therefore, a near zero torso velocity ($V_{torso}[n'] \approx 0$) is obtained.

The phase difference method is therefore a very simple method that can be used to estimate the velocity profile of human motion with less complexity. It is a simple method because the complexity associated with spectrogram estimation and the task of extracting the velocity profile from the spectrogram are avoided.

However, the phase difference method has its own drawbacks. The first drawback is its accuracy. As already mentioned, the phase difference method is dependent on the crude assumption that the reflection from the torso is the only significant reflection in the received signal. Therefore, the accuracy of this method is dependent on the ratio of the magnitude of the signal reflection from the torso to the magnitude of the aggregate received signal. The smaller this ratio, the less accurate the method will be. A detailed illustration on the accuracy of this phase difference computation is given in [2]. The second drawback of this method is that it gives inaccurate results when the background noise (the signal received when there is no motion) is not white. Such a coloured background signal may result from harmonics and other frequency components generated by imperfect transceivers. In presence of a coloured noise, the phase difference method gives a velocity estimate corresponding to the strongest background noise frequency. Therefore, unless background subtraction methods as suggested in [2] are used, the phase difference method does not estimate the velocity profile correctly in the absence of motion.

VII. Cadence Frequency Estimation

Cadence frequency is an important parameter of motion that shows how fast the appendages (legs and arms) of the body are oscillating. A cadence frequency spectrum shows the rate of change of each Doppler frequency: whether the magnitude of a particular Doppler frequency has a constant strength over time or has a certain rate of change. For instance, the torso has near to constant velocity (does not oscillate) as compared to the hands and legs whose velocity changes continuously in an oscillatory pattern. Such a pattern can be obtained from a cadence frequency spectrogram.

A cadence frequency spectrogram can be obtained by taking the FFT of the Doppler frequency versus time spectrogram over time at each Doppler frequency. Thus, the Doppler frequency versus time plot will be transformed into Doppler frequency versus cadence frequency plot. That is, the power of the signal $P_c[k, c]$ at a Doppler frequency index k and cadence frequency index c is given by:

$$P_c[k, c] = \left| \sum_{n'=1}^{N_w} |Y[k, n']| \, e^{-j \frac{2\pi}{N_w} c n'} \right|^2 \tag{16}$$

where $Y[k, n']$ is given by (7). The number of time windows involved in the FFT, N_w, should be short enough to estimate the change in cadence frequency pattern, *i.e.* to have enough time resolution, and it should be long enough to get enough cadence frequency resolution. Thus, an optimal window size should be taken considering these factors. The maximum cadence frequency to be considered depends on the time interval between consecutive windows.

Once the cadence frequency spectrogram is obtained, a simple method of summing the total power at each cadence frequency can be used to obtain the fundamental cadence frequency of the gait. Summing the powers at each cadence frequency over the Doppler bins gives a total power versus cadence frequency plot. The total power at a cadence frequency index c, $P_t[c]$, is thus given by:

$$P_t[c] = \sum_{k=1}^{N} P[k, c] \tag{17}$$

Based on the velocity profile model in (1), three peaks are expected on the cadence frequency plot. The first and strongest peak will be at a cadence frequency of 0 due to the near constant velocity of the torso, the second peak will be at the fundamental frequency ω_c and the third at the second harmonics $2\omega_c$. More harmonics orders might also be visible from the spectrogram. Therefore, the second peak from the cadence frequency plot is taken as the fundamental cadence of the gait.

VIII. Software Radio-based radar

The velocity profile and cadence frequency estimation methods discussed were evaluated in a set of experiments done using a GNU Radio-based active radar.

GNU Radio is an open source and free programming toolkit used for realizing software defined radios using readily-available, low-cost RF hardware and general purpose processors [27], [28]. The toolkit consists of a variety of signal processing blocks implemented in C++ that can be connected together using Python programming language. Some of the nice features of GNU Radio include the fact that it is free, open-source, re-configurable, can tune parameters in real-time and provides data flow abstraction. The Universal Software Radio Peripheral (USRP) is a general purpose programmable hardware that is commonly used as a front-end for GNU Radio [29].

The major components of the USRP are its FPGA, ADC/DAC sections and interpolating/decimating filters. The USRP is designed such that the high sampling rate signal processing, such as down conversion, up conversion, decimation, interpolation and filtering are done in the FPGA. The low speed signal processing such as symbol modulation/ demodulation, estimation and further signal processing takes place in the host processor. This lessens computational burden of the processor and makes signal processing easily manageable. The new USRP version, USRP2, has a Gigabit Ethernet interface

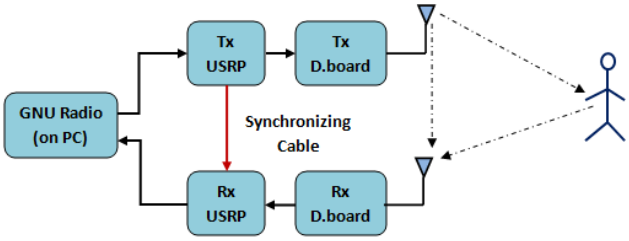

Figure 2. Monostatic radar setup using GNU Radio and USRP

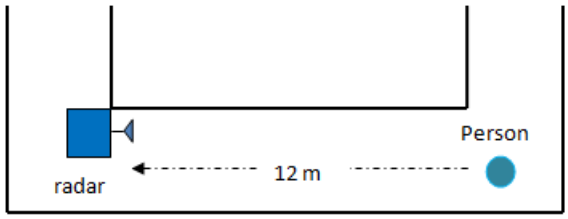

Figure 3. Walking experiment made in a corridor

allowing $25\,MHz$ RF bandwidth in and out of the USRP2 [27], [30].

GNU Radio and USRP have been widely used for prototyping in communication systems research [27]. Their adoption in a wide range of applications is motivated by the low cost, relative ease to use and flexibility. However, the use of USRP as a platform for building active radar is limited due to its low power and limited bandwidth. A possible design of USRP based long-range pulse radar is discussed in [31]. We instead used a USRP based continuous wave radar. To the best of our knowledge, our work is the first using USRP and GNU Radio as a short-range (indoor) active radar.

In our experiments, a USRP is used in conjunction with GNU Radio to implement a monostatic, unmodulated continuous wave radar. The USRP was equipped with a XCVR2450 daughterboard which works as the radar RF front-end in the $2.4 - 2.5$ and $4.9 - 5.9\,GHz$ bands. Figure 2 shows the schematics of our radar. The setup uses two separate USRPs, one for transmission and the other dedicated for reception. A cable between the boards ensures the two boards are synchronized to a common clock.

This radar platform is both low-cost and flexible. The carrier frequency, transmitter power, receiver gain, and other parameters are easily configurable in software.

IX. Evaluation

A detailed description of the different types of experiments done and the results obtained to evaluate the estimation of human movement parameters such as velocity profile, cadence frequency, displacement, activity index, direction of motion, etc., can be found in [32]. In this paper, only one of the experiments to evaluate the proposed velocity profile and cadence frequency estimation methods is described.

In the evaluation experiment, a person's movement in a confined area was measured using radar transmission frequency of $5\,\mathrm{GHz}$ and transmission power of $30\,\mathrm{dBm}$ (including antenna gains). The received signals were recorded in a data file and processed offline using MATLAB. The signal was low-pass filtered and decimated to a sampling rate Fs of $500\,S/s$. A window size of 100 samples which corresponds to $0.2\,s$ (where s represents seconds) is used assuming that the motion is piece-wise constant for a time duration of $0.2\,s$. An FFT size (N) of 500 and an overlap of 75% between the sliding windows are also used in the computation of both STFT and MUSIC spectrograms. In MUSIC, the autocorrelation lag parameter is set to $0.5L$ and the number of sinusoids D is set to 25. Such a value of D was chosen after experimenting on the received signal and taking into a account the discussion in Section IV.

Some important parameters of motion that can be easily observed from the spectrogram are discussed and compared with the actual motion of the subject. The velocity profile is estimated using the three methods discussed in Section VI. These velocity estimation methods are evaluated by computing the total distance covered based on the velocity profile estimated and comparing it with the actual distance covered by the subject which was measured manually. The weighted mean method is then selected to estimate and compare velocity estimations from the STFT and MUSIC based spectrograms. The number of steps taken to complete the motion are also recorded and used to evaluate the fundamental cadence frequency estimation method discussed in Section VII.

The experiment was done in a $2\ m$ wide and $12\ m$ long corridor as shown in Figure 3. The person stands at a distance of $12\ m$ in front of the radar for about $3\ s$ and starts walking towards the radar. Measurements with a timer and manual counting showed that it takes the person about $10s$ and 15 walking steps respectively to complete the $12\ m$ by walking.

A. Spectrograms

The STFT and MUSIC based spectrograms obtained from this experiment are shown in Figure 4 and 5 respectively. These spectrograms show the micro-Doppler pattern of the motion of the person over time. The following observations can be derived from these spectrograms:

- The time duration of motion recorded and the number of steps counted manually match the spectrogram pattern. The latter, which is counted to be 15 during the experiment, is equal to the number of spikes in the spectrogram (which is also 15 as Figure 4 shows more clearly). These spikes result from the forward swinging of the legs and arms. The periodic like pattern of the spikes in the spectrogram corresponds to the oscillation of the legs and arms that occur in a typical walking sequence. The spectrogram also shows that the backward swinging of the legs is small as compared to the forward swinging. This confirms the asymmetrical human movement model patterns observed in Figure 1.

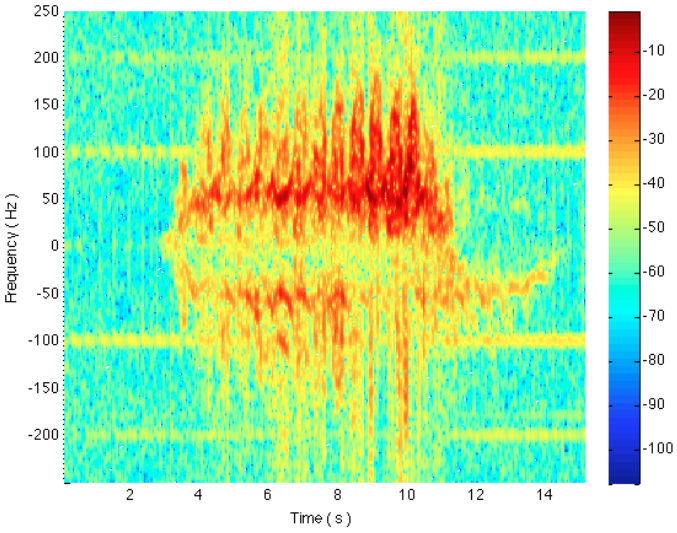

Figure 4. STFT based spectrogram estimate

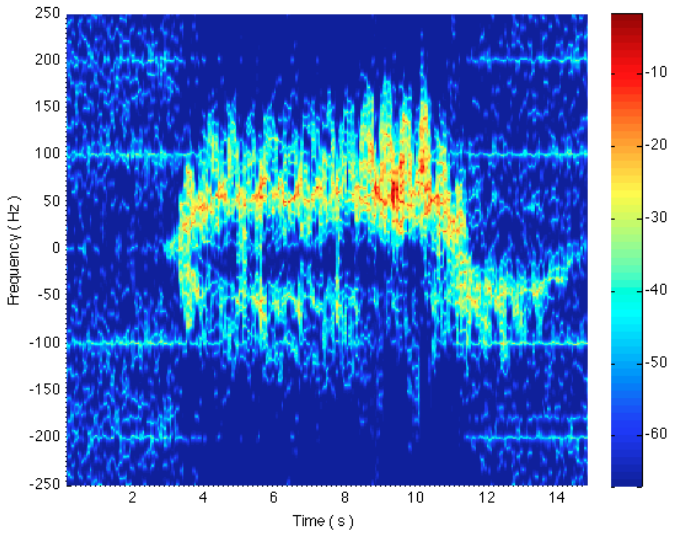

Figure 5. MUSIC based spectrogram estimate

- Even though the person is moving towards the radar which corresponds to a positive Doppler frequency, the spectrograms shows that there is an image micro-Doppler pattern of weaker power level in the negative Doppler frequencies. This confirms the image frequency problem discussed in Section VI.
- The STFT spectrogram has lower resolution than the MUSIC spectrogram as expected. On the other hand, the STFT micro-Doppler pattern is smooth as compared to a spiky MUSIC spectrogram that resolves the strongest frequencies as Figure 5 shows. Therefore, it can be deduced that the MUSIC spectrogram can be used to resolve the specific Doppler contribution of each of the rigid parts of the body.

B. Velocity Profile

The torso velocity profile estimated using the two spectrogram based velocity estimation methods, namely maximum power and weighted power methods, is shown in Figure 6. These estimates are based on the STFT spectrogram in figure 4. The performance of the phase difference method is also plotted in Figure 7 in comparison to the spectrogram based methods. The following can be said on the performance of these velocity profile estimation methods.

One possible measure to evaluate the accuracy of these methods is the total distance covered. This measure can only test the accuracy of the velocity profile estimations in average. To measure the total distance, a part of the spectrogram when the person is in motion must be considered (which is between $3\,s$ and $11\,s$ as shown in Figure 4). The total distance the person moved can then be estimated as the area under the velocity versus time curve. That is, Total distance $= 8\,s \cdot \sum_{t=3s}^{11s} V_{torso}\,[t]$. A total distance of $13.26\,m$ is obtained from the maximum power method which gives an error percentage of only 10% as compared to the manually measured distance of the corridor which is $12\,m$. Similarly, a total distance of $11.34\,m$ is obtained from the weighted mean method which gives an error percentage of only 5.5%. The total distance computed from the phase difference method is about $12.85\,m$ which results in an error percentage of 7%. These results show that all velocity profile estimation methods estimate the total distance with an error of less than 10% and the weighted mean method gives the best estimate.

The other measure that can be used is the performance of these methods when there is no motion (which is between $0\,s$ and $3\,s$ as shown in Figure 4). As Figure 7 shows, the maximum power method is able to perform well (outputs $V_{torso}[n'] = 0$) in absence of motion since it uses a threshold detector. On the other hand, the weighted power and phase difference methods have a significant error in the absence of motion. The figure shows that the phase difference method has the worst performance in the absence of motion due to the imperfect transceivers as discussed in Section VI-C. The background noise frequencies generated by our software radio-based radar prototype are evident from the horizontal symmetrical lines at $100\,Hz$ and $200\,Hz$ in Figure 4.

One of the nice properties of the weighted power method is that it is insensitive to symmetrical background noise. Therefore, the weighted power method has in average better accuracy than the phase difference and maximum power methods.

STFT versus MUSIC: The spectrograms in Figure 4 and 5 show that MUSIC is a good spectral estimator to resolve the contribution of the rigid parts of the body from the overall micro-Doppler signature. In order to evaluate the accuracy of velocity estimations computed from STFT and MUSIC spectrograms, the weighted power method is used. A comparative plot of the velocity estimations based on an STFT and MUSIC spectrogram is shown in Figure 8 for the duration of motion.

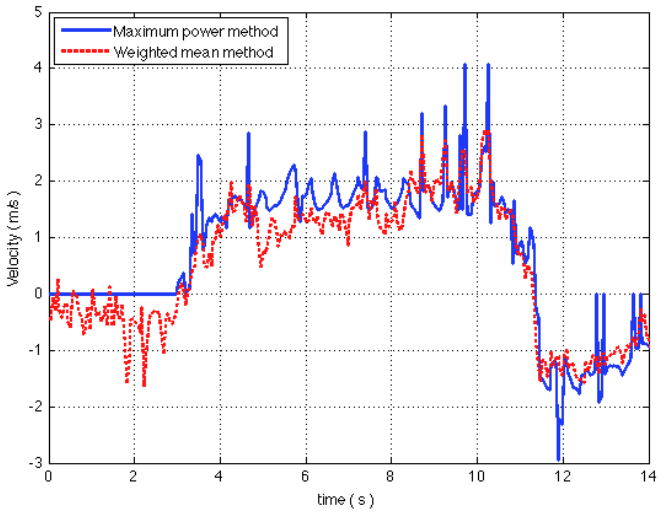

Figure 6. Spectrogram based velocity profile estimation methods

Figure 7. Phase difference method of velocity profile estimation compared with the spectrogram based estimates

The total distance is computed from these velocity estimations and is found to be $11.34\ m$ (estimation error of 5.5%) for the STFT based spectrogram and $12.34\ m$ (estimation error of 2.83%) for the MUSIC based spectrogram. This result suggests that the MUSIC based method outperforms the STFT based method in average. However, there is no significant difference between the two velocity profiles as Figure 8 shows. This is because the estimation methods in Section VI are not very sensitive to frequency resolution.

C. Cadence Frequency

The cadence frequency spectrogram can be obtained from the STFT or MUSIC spectrograms by applying Fourier transform at each Doppler frequency as discussed in Section VII. In this case the STFT spectrogram is used.

The cadence frequency spectrum obtained from the STFT spectrogram is shown in Figure 9. This spectrum shows the

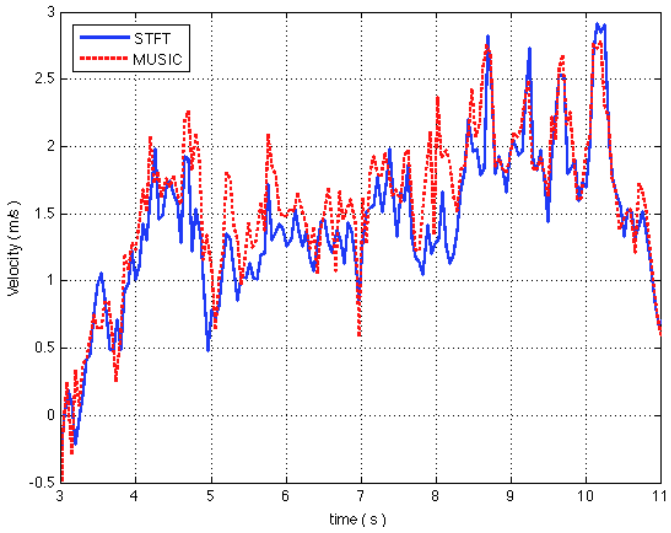

Figure 8. Velocity profile estimates using STFT and MUSIC based spectrograms

Doppler frequencies and their corresponding rate of oscillation contributed by the parts of the body. Small cadence frequency corresponds to no oscillation or variation of a Doppler component and large cadence shows high rate of oscillation. As indicated, the strongest Doppler frequency at zero cadence corresponds to the torso and the other strongest component at a higher cadence (which is the fundamental cadence of the gait) corresponds to the legs.

In order to obtain the fundamental cadence of the gait, the total power at each cadence frequency bin is summed and plotted as shown in Figure 10. This figure clearly shows two strongest cadence frequencies. It is evident from the human movement model in Section III that three strongest frequencies: 0, ω_c and $2\omega_c$ are expected from the cadence frequency plot. However, the second cadence is found to be weak here.

The fundamental cadence frequency (the second peak) is obtained from Figure 10 to be $1.74\,steps/s$. This parameter shows how many walking steps the person makes per second in average. As discussed in Section I, this parameter indicates the activity level and possibly the health status of a person. The cadence frequency estimation can be verified based on the manually recorded data when the experiment is done. It is stated that the number of steps the person took to cover the distance is 15 and the duration of motion as observed from the spectrograms to be $8\,s$. Therefore, the fundamental cadence frequency is $\frac{15\,steps}{8\,s} = 1.87\,steps/s$ which shows that the estimation results in an error of 6.9% only.

X. Conclusion

In this paper, pre-processing followed by STFT and MUSIC spectral estimators are applied to estimate the micro-Doppler signatures of human movement from a received radar signal. Elegant approaches to estimate the velocity profile and fundamental cadence frequency of motion are proposed.

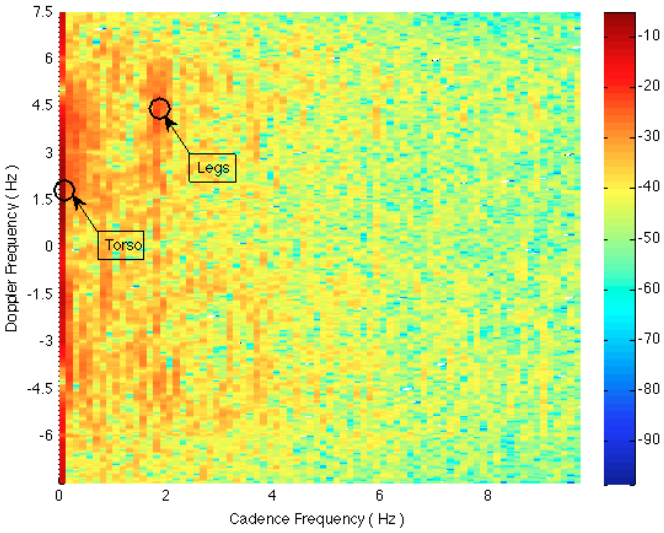

Figure 9. Cadence frequency spectrogram

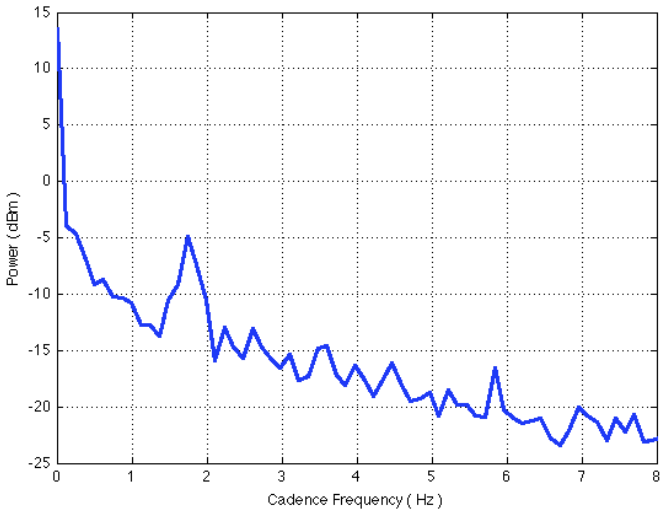

Figure 10. Total power versus cadence frequency showing the peak at the fundamental cadence frequency

Maximum power and weighted mean methods are suggested to extract the velocity profile from the spectrograms as well as an approximate but simple method based on phase difference computation. These velocity profile estimation methods are evaluated and compared against each other. A technique to extract the cadence frequency spectrum and the fundamental cadence frequency from the joint time-frequency estimation is also discussed and evaluated.

The maximum power, weighed mean and phase difference methods were able to measure the total distance covered with an error of 10%, 5.5% and 7% respectively. It is found that the maximum power method is error-prone since it needs a threshold and its performance depends on the choice and accurate estimation of the threshold value. The phase difference method is found to be accurate enough in the presence of motion. However, the sensitivity of this method to background noise makes it error-prone in the absence of motion. In weak image frequencies (outdoor environment for instance), the weighted power method is a suitable method. Its insensitivity to symmetrical coloured background noise is also another factor that makes the weighted mean method attractive. It can be concluded that the weighted power method outperforms both the maximum power and phase difference methods in average. However, the maximum power method is preferable in presence of strong image frequencies.

It is also shown that the MUSIC based spectrogram not only provides a resolved spectrogram showing the contribution of each component but also results in a smaller velocity profile estimation error. It is also found that the fundamental cadence frequency is estimated with an error of less than 7%. In general, it can be concluded that all velocity estimation methods suggested are able to estimate the velocity profile of human translational motion with an accuracy that is good enough for the applications concerned.

A major limitation of the velocity estimation methods discussed so far is that only the radial component of the velocity is being perceived and estimated by the radar. One way to achieve a better estimation is by combining information from two or more radars adjusted to monitor distinct directions. In addition, the velocity estimation methods discussed in this paper do not consider the possible presence of other interfering motions and assume that there is a single mover in the monitored environment. In applications where this is not acceptable, it is essential to be able to discriminate and track the velocity profiles of multi-movers. Research on extracting the velocity profile of multi-movers in indoor environment is considered in future work.

References

[1] B. Godana, G. Leus, and A. Barroso, "Estimating indoor walking velocity profile using a software radio-based radar," in *Sensor Device Technologies and Applications (SENSORDEVICES), 2010 First International Conference on*, pp. 44 –51, 2010.

[2] B. Godana, G. Leus, and A. Barroso, "Quantifying human indoor activity using a software radio-based radar," in *Sensor Device Technologies and Applications (SENSORDEVICES), 2010 First International Conference on*, pp. 38 –43, 2010.

[3] B. de Silva, A. Natarajan, M. Motani, and K.-C. Chua, "A real-time exercise feedback utility with body sensor networks," in *5th International Summer School and Symposium on Medical Devices and Biosensors, ISSS-MDBS 2008.*, pp. 49–52, June 2008.

[4] B. Lo, L. Atallah, O. Aziz, M. E. Elhew, A. Darzi, and G. zhong Yang, "Real-time pervasive monitoring for postoperative care," in *4th International Workshop on Wearable and Implantable Body Sensor Networks, BSN 2007*, pp. 122–127, 2007.

[5] S.-W. Lee, Y.-J. Kim, G.-S. Lee, B.-O. Cho, and N.-H. Lee, "A remote behavioral monitoring system for elders living alone," in *International Conference on Control, Automation and Systems, ICCAS '07.*, pp. 2725–2730, Oct. 2007.

[6] A. Purwar, D. D. Jeong, and W. Y. Chung, "Activity monitoring from real-time triaxial accelerometer data using sensor network," in *International Conference on Control, Automation and Systems, ICCAS '07.*, pp. 2402–2406, Oct. 2007.

[7] Z. Zhou, X. Chen, Y.-C. Chung, Z. He, T. Han, and J. Keller, "Video-based activity monitoring for indoor environments," in *IEEE International Symposium on Circuits and Systems, ISCAS 2009.*, pp. 1449–1452, May 2009.

[8] Y. Tsutsui, Y. Sakata, T. Tanaka, S. Kaneko, and M. Feng, "Human joint movement recognition by using ultrasound echo based on test feature classifier," in *IEEE Sensors*, pp. 1205–1208, Oct. 2007.

[9] S.-W. Lee, Y.-J. Kim, G.-S. Lee, B.-O. Cho, and N.-H. Lee, "A remote behavioral monitoring system for elders living alone," in *International Conference on Control, Automation and Systems, ICCAS '07.*, pp. 2725–2730, Oct. 2007.

[10] R. M. Narayanan, "Through wall radar imaging using UWB noise waveforms," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2008*, pp. 5185–5188, Mar. 2008.

[11] Y. Kim and H. Ling, "Human activity classification based on micro-Doppler signatures using an artificial neural network," in *IEEE Antennas and Propagation Society International Symposium, AP-S 2008*, pp. 1–4, July 2008.

[12] M. Otero, "Application of continuous wave radar for human gait recognition," in *Proc. SPIE*, vol. 5788, pp. 538–548, 2005.

[13] H. Burchett, "Advances in through wall Radar for search, rescue and security applications," in *Proc. Institution of Engineering and Technology Conference on Crime and Security*, pp. 511–525, June 13–14, 2006.

[14] D. J. Daniels, P. Curtis, and N. Hunt, "A high performance time domain UWB Radar design," in *Proc. IET Seminar on Wideband Receivers and Components*, pp. 1–4, May 7–7, 2008.

[15] S. S. Ram, Y. Li, A. Lin, and H. Ling, "Doppler-based detection and tracking of humans in indoor environments," *Journal of the Franklin Institute*, vol. 345, pp. 679–699, 2008.

[16] M. G. Anderson, *Design of multiple frequency continuous wave radar hardware and micro-Doppler based detection and classification algorithms*. PhD thesis, University of Texas, Austin, May 2008.

[17] S. Gurbuz, W. Melvin, and D. Williams, "Detection and identification of human targets in Radar data," in *Proc. of SPIE Defense and Security Symposium*, 2007.

[18] C. Hornsteiner and J. Detlefsen, "Characterisation of human gait using a continuous-wave Radar at 24 GHz," *Advances in Radio Science*, vol. 6, pp. 67–70, 2008.

[19] J. Geisheimer, W. Marshall, and E. Greneker, "A continuous-wave (CW) radar for gait analysis," in *Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 1, pp. 834 –838 vol.1, 2001.

[20] C.-P. Lai, Q. Ruan, and R. M. Narayanan, "Hilbert-Huang transform (HHT) processing of through-wall noise radar data for human activity characterization," in *IEEE Workshop on Signal Processing Applications for Public Security and Forensics, SAFE'07*, pp. 1 –6, April 2007.

[21] L. Du, J. Li, P. Stoica, H. Ling, and S. Ram, "Doppler spectrogram analysis of human gait via iterative adaptive approach," *Electronics Letters*, vol. 45, pp. 186 –188, 29 2009.

[22] R. Boulic, N. Magnenat-thalmann, and D. Thalmann, "A global human walking model with real-time kinematic personification," *The Visual Computer*, vol. 6, pp. 344–358, 1990.

[23] Z. Zhang and N. Troje, "3D Periodic Human Motion Reconstruction from 2D Motion Sequences," in *Computer Vision and Pattern Recognition Workshop, CVPRW '04*, pp. 186–186, June 2004.

[24] M. Skolnik, *Introduction to Radar Systems*. McGraw-Hill, 2002.

[25] J. Geisheimer, E. Greneker, and W. Marshall, "High-resolution Doppler model of the human gait," in *Proceedings of SPIE*, vol. 4744, 2002.

[26] P. Stoica, *Introduction to spectral analysis*. Prentice Hall, 1997.

[27] "GNU Radio." http://gnuradio.org/redmine/wiki/gnuradio. Last accessed on January 2011.

[28] N. Manicka, "GNU radio testbed," Master's thesis, University of Delaware, 2007.

[29] M. Ettus, *USRP users and developer's guide*. Ettus Research LLC.

[30] "Universal Software Radio Peripheral." http://www.ettus.com/. Last accessed on January 2011.

[31] L. K. Patton, "A GNU radio based software-defined radar," Master's thesis, Wright State University, 2007.

[32] B. Godana, "Human Movement Characterization in Indoor Environment using GNU Radio Based Radar," Master's thesis, Delft University of Technology, 2009.

Measurement-Based Performance and Admission Control in Wireless Sensor Networks

Ibrahim Orhan
School of Technology and Health
KTH
Stockholm, Sweden
Ibrahim.Orhan@sth.kth.se

Thomas Lindh
School of Technology and Health
KTH
Stockholm, Sweden
Thomas.Lindh@sth.kth.se

***Abstract*—This journal paper presents a measurement-based performance management system for contention-based wireless sensor networks. Its main features are admission and performance control based on measurement data from light-weight performance meters in the endpoints. Test results show that admission and performance control improve the predictability and level of performance. The system can also be used as a tool for dimensioning and configuration of services in wireless sensor networks. Among the rapidly emerging services in wireless sensor networks we focus on healthcare applications.**

Keywords - wireless sensor network, admission control, performance monitoring and control.

I. INTRODUCTION

Wireless personal area networks have emerged as an important communication infrastructure in areas such as at-home healthcare and home automation, independent living and assistive technology, as well as sports and wellness. Initiatives towards interoperability and standardization are taken by several players e.g., in healthcare services. Zigbee Alliance has launched a profile for "Zigbee wireless sensor applications for health, wellness and fitness" [2]. The Continua Health Alliance promotes "an interoperable personal healthcare ecosystem" [3], and at-home health monitoring is also discussed in an informational Internet draft [4]. It shows that wireless personal area networks, including body sensor networks, are becoming more mature and are considered to be a realistic alternative as communication infrastructure for demanding services. However, to transmit data from e.g., an ECG in wireless networks is also a challenge, especially if multiple sensors compete for access as in CSMA/CA. Contention-based systems offer simplicity and utilization advantages, but the drawback is lack of predictable performance. Recipients of data sent in wireless sensor networks need to know whether they can trust the information or not. To address this problem we have developed a performance meter that can measure the performance [5], and furthermore, feed a performance control system with real-time measurement data [6]. This paper also discusses whether admission control in combination with a system for continuous performance management can provide improved and more predictable performance. Admission control is used in many traditional telecom systems. It is also proposed in new Internet service architectures [7] to provide guarantees for quality of service. In this paper we present a method for measurement-based admission control in wireless personal area sensor networks for contention-based access. It is implemented as a part of an integrated performance management system that comprises performance monitoring, admission control and performance control.

The rest of the paper is organized as follows: a survey of related work in Section II; performance management in wireless sensor networks in Section III; measurement-based performance and admission control in Section IV; use cases and test results in Section V; and finally the conclusions in Section VI. This journal paper is an extension of a paper on admission control presented at a conference [1]. It provides a more detailed view of the other parts of the system, as well as the entire system for performance management.

II. RELATED WORK

Performance in contention-based wireless networks using CSMA/CA has been studied extensively. Measurements, simulations and theoretical studies show that the loss ratio increases with the traffic load and number of sending nodes. Bianchi [8] has derived an analytical Markov chain model for saturated networks, further developed in [9] and extended to non-saturated networks in [10]. Channel errors due to e.g., external disturbances and obstacles in the environment, can of course increase the loss ratio further. Another related problem, studied in [11], is the reduced throughput in multi-hop networks, with one or several intermediate nodes between sender and receiver. Dunkels and Österlind [11] found that the implementation of packet copying in intermediate forwarding nodes has significant impact on the throughput.

Performance in low-rate WPAN has been analyzed in several simulation studies ([12] and [13]). A performance meter that keeps track of losses, inter-arrival jitter and throughput has been developed [5]. Several papers have also addressed congestion and rate control in WLAN and LR-WPAN. CODA (congestion detection and avoidance in sensor networks) is a control scheme that uses an open-loop backpressure mechanism as well as a closed-loop control, where a sink node can regulate a source node's sending rate by varying the rate of acknowledgements sent to the source [14]. CARA (collision-aware rate adaptation) uses the RTS packets in IEEE 802.11 as probes to determine whether losses are caused by collisions (related to CSMA/CA) or by channel errors [15].

Our implementation of admission control, to accept or reject a request to join the network, is based on measurements of performance parameters, mainly the packet loss ratio. A

similar probe-based admission control procedure has been suggested for differentiated Internet services [7]. Alternatively, one can measure the available capacity between two endpoints, or on specific links in a network. Pathrate, Pathload and BART are examples of implementations of such estimation tools ([16], [17] and [18]). SenProbe [19] estimates the maximum achievable rate between two endpoints in wireless sensor networks by injecting packet trains and analyze the dispersion between the packets. Some experimental studies indicate that measurements of available capacity in wireless networks often are inaccurate, especially for multiple hops [20]. Instead of active measurements, the contention-aware admission control protocol (CACP) estimates the available capacity by letting each node measure the amount of time the channel is busy [21]. Perceptive admission control (PAC) is an extension of CACP to encompass node mobility [22]. We have preferred a straightforward approach where the decision to either accept or reject an admission request is based on direct measurements and estimates of the performance parameters that are decisive for the quality of services.

III. Performance Management in Wireless Sensor Networks

A network scenario for the performance management system in this paper is depicted in Fig. 1. It consists of wearable sensors, such as ECGs, accelerometers, pulse-oximeters, fixed environment sensors, a coordinator, and intermediate nodes with routing and forwarding capabilities. An application program, running in the coordinator, processes sensor data from the sources and sends the information along with an estimate of the transmission quality to the remote end-user application for presentation and storage. The transmission quality can be expressed in terms of e.g., the statistical uncertainty of estimated parameters and the highest frequency component in a signal to be recovered by the receiver.

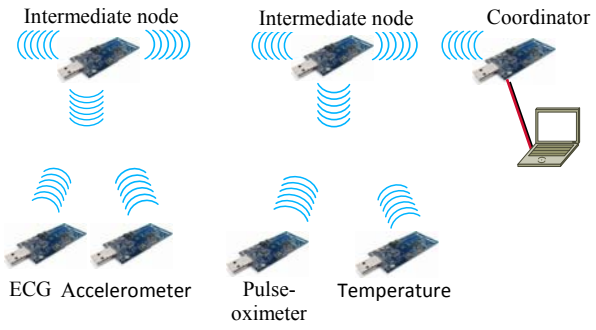

Figure 1. A network scenario where the performance management system is implemented in the coordinator and source nodes.

The performance monitoring and control capabilities can be implemented as add-on functions to be used by applications running in the communicating endpoints, e.g., sensor nodes and a coordinator, and not link by link. The ambition has also been to minimize the traffic overhead and energy consumption. The system is targeted to wireless sensor networks that use contention-based access, but can of course also be used in combination with contention-free access, such as guaranteed time slots. The applications, e.g., streaming data from accelerometers and ECGs, require certain levels of throughput and a low loss ratio, however not necessarily zero. The aim is, firstly, to provide quality estimates of the transmitted parameters, and secondly, to reuse this information for admission and performance control of information loss, delays and throughput. This closes the loop between measurements and control.

Admission control needs to be seen in the context of other necessary functions, especially performance measurements and control. The performance manager consists of the following functions: a performance meter that collects measurement data; admission control that handles requests to join the network; and performance control that maintains the quality of service for the admitted sensor nodes. The performance meter provides feedback information for admission and performance control. Fig. 2 shows the relationship between these functions. A request from a sensor node to join the network is handled by the admission control based on feedback from the meter. The performance control function is responsible for maintaining the desired quality-of-service once the sensor nodes are allowed to use the wireless channel. The performance meter is described in the following subsection (III.A) and admission and performance control in Section IV.

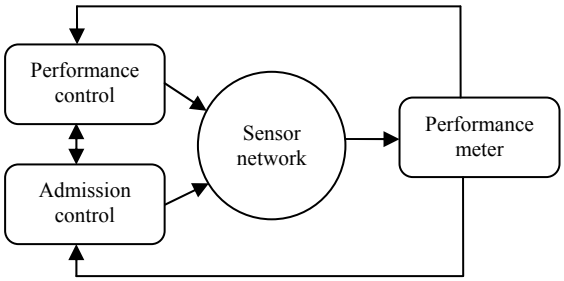

Figure 2. The performance manager consists of performance control and admission control. The performance meter supports the manager with measurement data.

A. Performance Meter

The approach is to combine active and passive techniques, inspired by the results from measurements in wired networks ([23] and [24]). A light-weight performance meter is implemented in each node. The meter consists of two counters that keep track of the number of sent and received packets and bytes, and a function that can inject monitoring packets. These dedicated measurement packets are inserted between blocks of ordinary data packets as seen in Fig. 3. They contain a sequence number, a timestamp and the cumulative number of packets and bytes transmitted from the sending node to the receiving node.

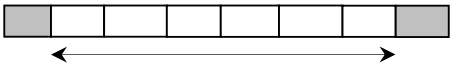

Figure 3. A monitoring block surrounded by two monitoring packets.

The interval between the monitoring packets, i.e. the size of the monitoring block, can be expressed in number of packets or a time interval, constant or varying randomly around a mean value. When a monitoring packet arrives, the receiving node stores a timestamp and the current cumulative counter values of the number of received packets and bytes from the sending node. Observe that for *n* sending nodes, the receiving node maintains *n* separate monitoring functions, one for each sending node.

Synchronization of the clocks in the participating nodes is not required. The local timestamps are used to calculate the inter-sending and inter-arrival times between pairs of monitoring packets. The inter-arrival jitter can then be calculated in a similar way as for RTP timestamps [25]. This means that the arrival time variation is estimated based on the monitoring packets, which represent samples of the ordinary data packet inter-arrival variation. Packet loss, on the other hand, is measured passively and directly using the counters.

1) Performance metrics

The following metrics can be calculated and estimated based on the collected measurements described in the previous subsection.

- Packet loss ratio: long-term average and average per monitoring block.
- The length of loss and loss-free periods defined as the number of consecutive monitoring blocks with or without losses. Can be expressed in time units, number of blocks, or number of packets and bytes.
- Inter-arrival jitter, J, is defined as $J=(r_n-r_{n-1})-(s_n-s_{n-1})$, where s is the sending time and r is the receiving time. The monitoring packets provide samples of this delay variation metric, which means that the uncertainty of the estimated statistics (mean value, median, percentiles etc.) is determined by the number of samples, and the variance of the delay process.
- Data throughput between sender and receiver can be calculated as a long-term average and also per monitoring block. The resolution of the peak rate is determined by the ratio between monitoring packets and ordinary data packets. This can also be seen as a measure of utilized capacity.

2) Meter and monitoring packet implementation

The performance meter is programmed in nesC [26] for TinyOS 2.1. The sensor nodes read samples from the sensors (ECG, accelerometer and temperature), assemble the samples and send them in packets to the coordinator (Fig. 4). The number of bytes and packets are counted. The cumulative number of bytes and packet and a timestamp are inserted into a monitoring packet, which is sent after every *n* ordinary data packet. A monitoring packet is 17 bytes long and includes the following fields: a start flag, a timestamp when packet is sent, type, a sequence number, number of packets sent, number of bytes sent, and a stop flag. The flags enable the coordinator to distinguish a monitoring packet from ordinary data packets. The sequence numbers identify and keep track of the monitoring packets. The packet and byte fields contain the cumulative number of bytes and packets sent. Finally, the type field enables measuring several sensor data flows from the same node.

Each time the coordinator receives a data packet, it updates the number of bytes and packets received from each sensor. The coordinator uses the source field in the CC2420 radio header to distinguish the packets from different sources. When the coordinator receives a monitoring packet, it stores a timestamp and the cumulative counter values of the number of received packets and bytes from the sending node. Fig. 4 shows the measurement data sent from the performance meter to the performance manager. The table in the lower left part of Fig. 4 shows the information in each monitoring packet sent from a sensor node: a timestamp, the total number of bytes and the total number of packets sent from the sensor node. The table to the right shows the corresponding information added by the coordinator for each received monitoring packet.

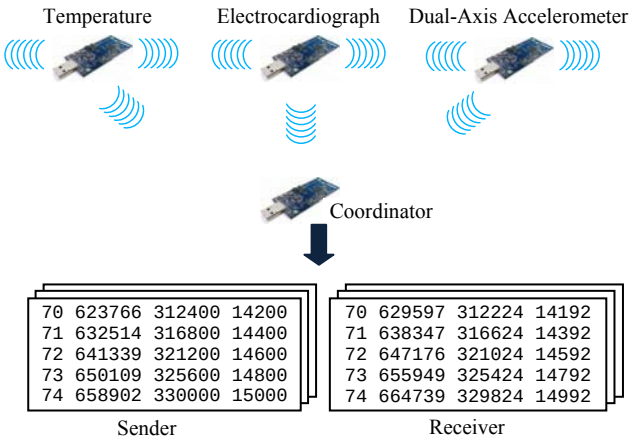

Figure 4. Measurement data from the sender and the receiver nodes. Columns from left to right: monitoring packet sequence no, timestamp (ms), cumulative number of bytes and packets.

IV. Admission and Performance Control in Wireless Sensor Networks

In this section the main idea behind the admission control (Section IV-A) and performance control (Section IV-B) system is presented. A star topology network controlled by a coordinator is used.

A. Measurement-Based Admission Control for Contention-Based Access

A typical application scenario is healthcare at-home with a number of sensors, such as ECGs, pulse-oximeters, accelerometers etc., connected to a coordinator. Fig. 5 shows a scenario with three sensor nodes connected to a coordinator sharing the same wireless channel that applies the CSMA/CA access method. Several hops between the sensor nodes and the coordinator, as well as mobile sensor nodes, are also a feasible scenario. Sensor node A and sensor node B in Fig. 5 are already connected to the wireless channel transmitting sensor data to the coordinator. The sensor nodes have a specified throughput and an upper limit for the packet loss ratio. Sensor node C requests admission to join the network for a specified throughput and packet loss ratio.

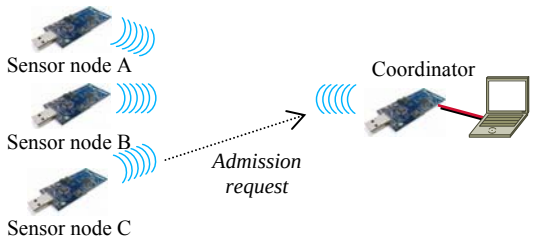

Figure 5. Sensor node C requests to share the wireless channel, already used by sensor node A and sensor node B.

The idea behind admission control is to accept or reject new sensor nodes to an existing network, while protecting the performance of already admitted nodes. Our purpose is to study whether it is feasible or not to use admission control in contention-based wireless sensor networks. The approach is to found the decision, to accept or reject an admission request, on estimates of real-time measurement data provided by a performance meter. A sensor node that intends to enter the network specifies the sampling rate, the sample size and the performance requirements. The verdict, to accept or reject the request, is determined by the outcome of probe packets transmitted during a test period. The probe packets sent from the requesting node to the coordinator should be of the same kind as the ordinary traffic it will transmit if admitted. The exchanged messages between a requesting node and the coordinator are described in the next subsection.

Strict performance guarantees are not feasible in contention-based access networks. However, many applications do not require completely loss-free transmission and are satisfied with soft performance requirements e.g., upper limits on packet loss and delay variation. The need for performance guarantees and predictability in contention-based networks for such applications is addressed in one of the use cases in Section V-C.

1) Messages between the coordinator and sensor nodes

A simple protocol for exchange of messages between the coordinator and the sensor nodes have been defined (Fig. 6). Sensor nodes send requests to join the network for a specified sampling rate, sample size and upper limits on performance parameters. If the coordinator is not busy handling previous requests, it will approve further processing. The sensor node is then instructed to start transmitting probe packets interleaved by monitoring packets. When the test period ends, the sensor node asks the coordinator for the decision. Having received 'accept', the sensor node begins transmitting its ordinary data packets to the coordinator. Monitoring packets are inserted between blocks of *n* data packets or with certain time intervals, to provide the performance meter in the coordinator with real-time updates of the transmission quality.

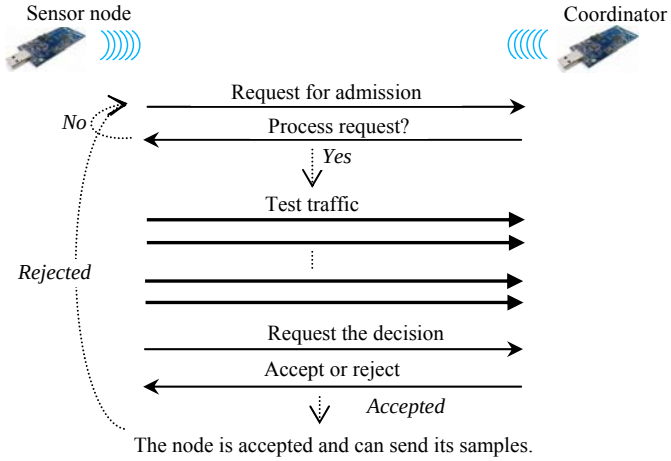

Figure 6. The messages between a sensor node and the coordinator during the admission phase. The arrows in thin lines are signalling messages and the arrows in bold lines represent probe packets in the test traffic phase.

2) Admission test period

The sensor nodes transmit probe packets during the test period in the same way as they intend to do if the request is accepted. The performance meter will report performance data for traffic between the coordinator and all sensor nodes as well as the test traffic from the requesting sensor node. Admission is accepted if the averages of the performance parameters for any of the already permitted nodes, including the requesting node, are below the threshold value. Admission can be denied to protect the existing nodes from performance degradation. The length of the test period is a trade-off between retrieving enough information from the probe packets and minimizing the effect on the other sensor nodes' performance. The first priority is to protect the already admitted nodes. The test traffic phase will be interrupted as soon as the probe packets have the effect that e.g., the loss ratio threshold is exceeded. The probe packets sent during the test period can be seen as a sampling process of the wireless channel, where the outcome of each sampling event is that the packet is lost or succeeds. The probability to lose a packet depends on the total traffic load and the number of nodes that are transmitting (ignoring radio channel disturbances). The number of samples needed for a given confidence level is determined by the variance of the traffic load. We have assumed that the sampling frequencies of the sensors are stable. This is a reasonable assumption for the kind of the applications the system is intended for. It means that the variance of the traffic load over time is low, and accordingly, that the number of probe packets can be kept small. The experiences from the use cases (Section V-C) in a normal home environment confirm that a test period of less than 30 seconds is sufficient. The length of the test period is further discussed in Section V-C.

B. Performance Control System

The aim of the performance control system is, firstly, to provide quality estimates of the transmitted parameters, and secondly, to reuse this information for systems management and enable performance control in real-time e.g., to minimize

information loss and maintain a desired throughput. The output of the performance control system can also be to change the transmission power, enable or disable acknowledgement, etc. Applications, such as streaming data from accelerometers and ECGs in Fig. 7, require certain levels of throughput and a low loss ratio, however not necessarily zero.

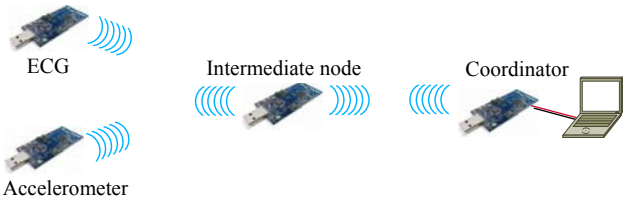

Figure 7. Two sensor nodes transmitting data to a coordinator via an intermediate node.

The performance control system (Fig. 8), implemented in a coordinator node, bases its decisions on the feedback information it receives from the meter e.g., packet loss, delays and throughput (packet loss in used in our cases). The meter delivers these performance updates for each incoming monitoring block e.g., once a second. The performance monitoring and control method has three main parameters. Firstly, the size of the monitoring block that determines the resolution of the performance metrics as well as the response time for the control actions. Secondly, the number of previous monitoring blocks (B_n, B_{n-1}, B_{n-2} etc), and their relative weight. The performance measurement results are calculated per each received monitoring block. To which degree the control method can rapidly adapt to changes is determined by these parameters. Thirdly, a step size (Δt) controls the time interval between transmitted packets (and thereby the packet frequency). This step size determines the response time and also the stability of the system.

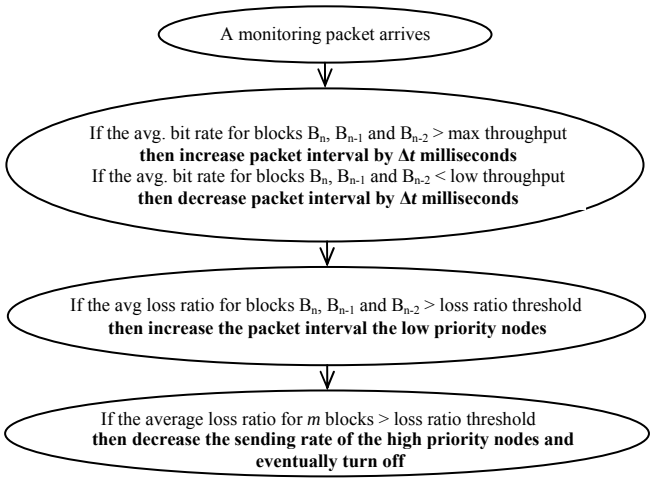

Figure 8. Flow diagram of the implemented control algorithm for prioritized nodes.

1) Algorithm to control throughput and loss ratio

The output of the control algorithm, to decrease or increase the packet frequency, is based on performance data from the current and previous monitoring blocks. The loss ratio and throughput (received bits per second) for a number of the recently received monitoring blocks are kept in memory. The manager sends a request message to a sensor node to either reduce or increase the packet frequency by adding (or subtracting) Δt milliseconds to (or from) the time interval between the transmitted packets.

2) Control algorithm with priority

A performance control system can support quality-of-service by assigning different priority to sensor nodes. Performance control with priority is primarily based on feedback information regarding packet loss and throughput from the respective source nodes. A use case with two levels of priority is described in more detail in Section V-B. High priority means that the required throughput (received bits per second) is maintained and the packet loss ratio is kept below a threshold for the prioritized nodes, possibly at the expense of nodes with low priority. If the loss ratio for the high-priority node is above the threshold, the manager will instruct the low-priority sensor nodes, to decrease their transmission rate step by step until the loss ratio for the high-priority node is below the threshold. If the loss ratio still is above the threshold, the sending rate of the high-priority nodes will be decreased as well, and eventually turned off if the loss ratio remains too high.

V. USE CASES

In this section, we present use cases where the performance meter, performance control and admission control is used. Section V-A shows how the performance meter is used for online transmission quality feedback. Examples of parameters and statistical uncertainty are presented. Section V-B contains two cases: performance control to maintain throughput and keep packet loss below a threshold; and control with different and dynamically assigned priority. Section V-C illustrates the potential performance problems with contention-based access and the need for admission control, as well as continuing performance monitoring and control. The sensor node platform TmoteSky [28], running TinyOS 2.1 and programmed in nesC, is used in all cases below. The radio (CC2420) and link layer are compliant with IEEE 802.15.4 LR-WPAN [27] in contention-based access mode.

A. Performance Meter – Online Transmission Quality Feedback

1) The testbed and measurement scenarios

Two different network scenarios are studied. In Fig. 9 the sensor nodes are attached to the coordinator in a star topology.

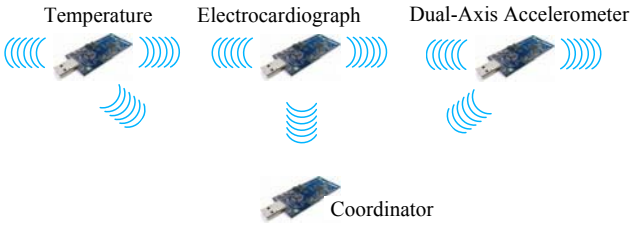

Figure 9. A network scenario with three sensor nodes and a coordinator.

In the second scenario (Fig.10), a sensor node is placed two hops away from the coordinator. The intermediate node forwards the packets from the sensor node to the coordinator. The buffer size in the intermediate node is 20 packets. In both scenarios, samples of sensor data are sent from the sensor node to the coordinator.

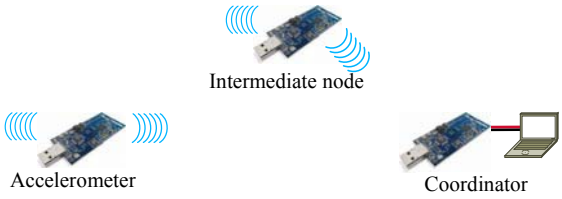

Figure 10. A network scenario with two hops between the sensor node and coordinator.

a) *The temperature sensor*

The temperature sensor in Fig. 9 is sampled twice a second and the collected samples are sent immediately to the coordinator. Monitoring packets are inserted between blocks of 100 data packets (6 byte payload).

b) *The ECG sensor*

One of the sensors in Fig.9 reads samples at 200Hz from the ADC-12 (analog-to-digital converters, 12 bits resolution) connected to an ECG. The samples are collected during five seconds. The radio is switched on, the samples are sent to the coordinator, and then switched off. This procedure is then repeated. Each packet contains 13 samples. The idea is to keep the radio turned off as long as possible and send several samples in each packet, in order to minimize the power consumption. In this case, the sensor node transmits 77 packets back-to-back every five seconds. A monitoring packet is inserted between blocks of approximately 100 ordinary data packets.

c) *The dual-axis accelerometer sensor*

A multi-sensor board (SBT80 from Easysen [29]) with a dual-axis accelerometer sensor is connected to two ADCs, one ADC for each axis. The accelerometer is sampled at 100Hz. The radio is only turned on during transmission. The sensor node sends 20 packets per second to the coordinator. Each packet carries 10 samples, 5 samples from each axis. Monitoring packets are inserted between blocks of 200 data packets, i.e. with approximately 10 seconds intervals.

2) Results and discussion

In this section some results using the performance meter in the two scenarios in Fig. 9and Fig.10 are presented.

a) *Loss periods and loss-free periods*

The loss ratio per monitoring block during the measurement period for the accelerometer data is illustrated in Fig. 11. The distinct loss events in the beginning of the measurement period are caused by radio interferences. Table I shows the loss ratio per monitoring block and the mean length of loss periods and loss-free periods for the three wireless links in Fig. 9.

TABLE I. PACKET LOSS RATIO BETWEEN SENSOR NODES AND COORDINATOR

	Acc–Coord.	ECG–Coord.	Temp-Coord.
Mean loss ratio	0.038	0.002	0.006
Max loss ratio	0.935	0.040	0.100
Min loss ratio	0.000	0.000	0.000
Loss period mean length (s)	37s	6s	11s
Loss-free period mean length (s)	15s	40s	165s

The loss ratio during the three loss periods is between 0.8 and 0.9 (Fig. 11). The length of the loss-periods (consecutive monitoring blocks that contain at least one lost packet) is shown in Fig. 12.

b) *Inter-arrival delay variation*

Table II shows the inter-arrival delay variation (jitter) for the scenario in Fig. 10 with two hops between the sensor node and the coordinator compared to one hop. The sensor node transmits 20 packets per second. The radio communication is not exposed to disturbances in this case. Fig. 13 and Fig. 14 show that the inter-arrival jitter is several times higher with an intermediate node then without it. Packet loss for two hops is also considerably higher compared to one hop. The high levels of inter-arrival jitter and packet loss in the two-hop case is due to the intermediate node's receiving and forwarding capabilities.

TABLE II. INTER-ARRIVAL JITTER (MS)

Inter-arrival jitter (ms)	One hop	Two hops
Maximum	13	59
Minimum	0	0
Standard deviation	4.0	12.5

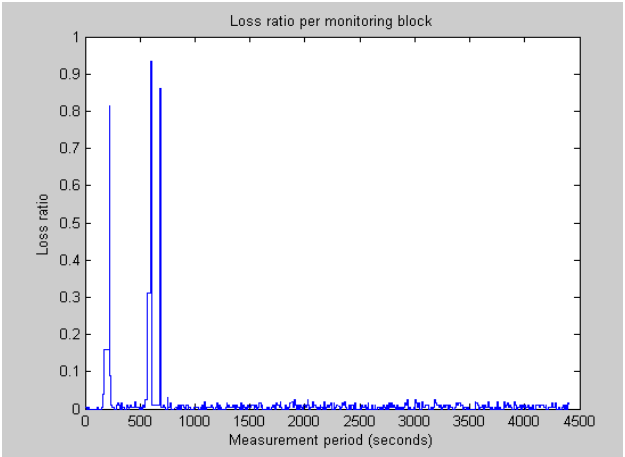

Figure 11. The loss ratio per monitoring block for accelerometer data in Fig. 10.

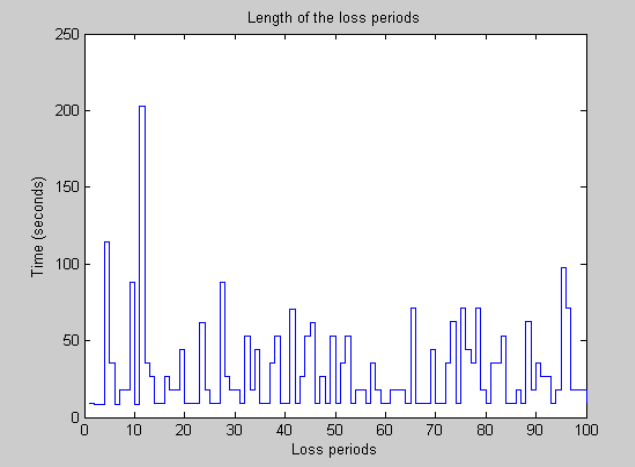

Figure 12. The length of loss periods in seconds.

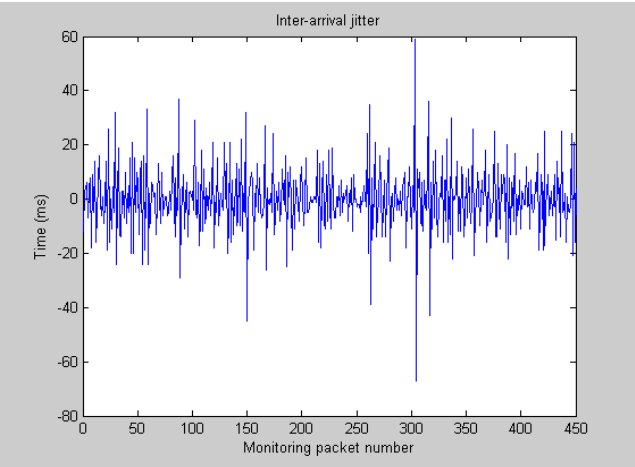

Figure 13. Inter-arrival jitter for a two-hop case.

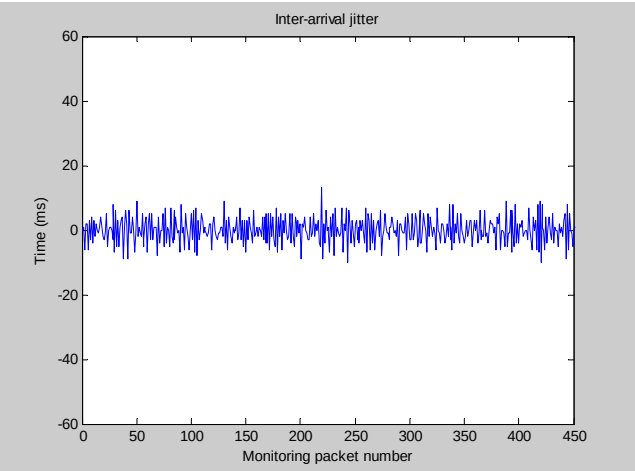

Figure 14. Inter-arrival jitter for a one-hop case.

c) *Uncertainty in parameter estimation*

The result from the performance meter can be used for calculating the statistical uncertainty of the parameter estimates based on samples from sensors. Table III shows how the confidence interval increases and highest frequency component in a received signal decreases when the loss ratio increases due to network performance degradation.

TABLE III. UNCERTAINTY IN ESTIMATION OF LOSS RATIO

Monitoring block duration (s)	Loss ratio	Conf. interval (0.99 level, stdev=4)	Highest frequency component in received signal
10s	0.025	0.93	50Hz
20s	0.313	1.11	35Hz
40s	0.935	3.62	3Hz
10s	0.010	0.93	50Hz
80s	0.861	2.47	18Hz
10s	0.030	0.94	50Hz
10s	0.005	0.93	50Hz
10s	0.005	0.93	50Hz
10s	0.010	0.93	50Hz

Details from the longest loss period in Fig. 12 are shown in Table III. The entire loss period consists of 20 monitoring blocks and lasts for 200 seconds. The monitoring block size in this case is 10 seconds. However, several blocks are longer e.g., the second, third and fifth row in Table III. The explanation is that monitoring packets, as well as data packets, may disappear before they arrive at the destination during a loss period. If one or several monitoring packets in a row are lost, the original monitoring blocks are merged into a larger block. Row 5 in Table III is a concatenation of 8 original blocks, where 7 monitoring packets were lost.

The loss ratio in Table III stretches from 0.005 to 0.935. The increased statistical uncertainty in estimating the mean value as the losses increase is shown in the third column. The standard deviation is around 4 units and the confidence level is chosen to be 0.99. The resulting confidence interval for an ideal communication channel without losses will be 0.92. A loss ratio of 0.313 (second row) leads to a confidence interval of 1.11, and a loss ratio of 0.935 gives a four times wider confidence interval (3.62). The number of samples, n, for a certain confidence interval, d, and confidence level (z=2.58 for 0.99 confidence level), and standard deviation, s, is given by, $n = \frac{z^2 \cdot s^2}{(d/2)^2}$.

The highest frequency component that can be recovered by the receiver for a 100Hz sampling rate is 50Hz (the sampling theorem). In this case the actual highest frequency component in the received signal is as low as 3Hz during the 36 seconds long period with a loss ratio of 0.935.

B. Performance Control Algorithms

Three examples of performance control are presented in this section. The purpose of the first control algorithm is to maintain throughput and minimize losses for a node with high priority by punishing nodes with low priority (Section V-B.1). In the second case, where all nodes have the same priority (Section V-B.2), each node tries to maximize its throughput under the condition that the loss ratio is below a threshold. The third case (Section V-B.3) is a combination of the two previous ones. From the beginning both nodes have the same priority. After a certain time, one of the nodes is

dynamically assigned high priority and higher throughput. Finally, we show how the number of hops between a sensor node and the receiving coordinator determine the end-to-end throughput (Section V-B.4).

Fig. 15 shows the network scenario for the first case with two sensor nodes that are streaming ECG samples and accelerometer samples to the coordinator through a forwarding intermediate node.

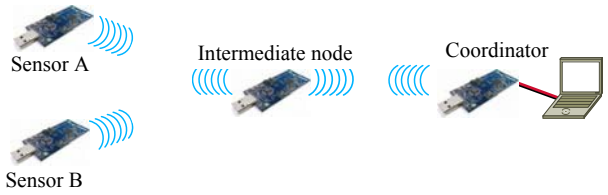

Figure 15. Two sensor nodes transmitting data to a coordinator via an intermediate node. Sensor node A has high priority and sensor node B has low priority.

1) Control with priority

The control algorithm in this case means that one of the sensor nodes is assigned high priority. The goal is to maintain throughput and keep loss ratio below an upper limit (0.02) for the high-priority node. The loss ratio threshold is computed as a weighted average of the three recent consecutive monitoring blocks and compared to the threshold 0.02. The required bit rate is 8kb/s, which corresponds to approximately 250Hz sampling rate per axis for a two-axis accelerometer or a 500Hz ECG.

Fig. 16 to Fig. 19 illustrate how the implemented algorithm works in practice. The high-priority node starts from 10kb/s and slows down to the expected bit rate 8kb/s (Fig. 16). The second node is turned on shortly thereafter (t≈80s) at a rate of nearly 16kb/s (Fig. 17). The received bit rate from the high-priority node falls sharply (Fig. 16). The solid lines (blue) show the received bit rate measured at the coordinator. The dotted lines (red) represent the sending bit rate from the sensor node. The loss ratio for the high-priority node peaks at almost 0.45 (Fig. 18), when the second node starts transmitting. The loss ratio for the low-priority nodes is shown in Fig. 18.

The performance manager reads the performance data provided by the meter for each block of incoming data packets. The monitoring block size is 100 packets in this test case. As soon as the manager detects the increased loss ratio for the high-priority node, it will instruct the other node to slow down. The low-priority node will directly decrease the transmitting rate (Fig. 17), which results in lower loss ratio (Fig. 18) and higher throughput (Fig. 15) for the prioritized node. As the loss ratio approaches the threshold, the sending rate of the low-priority node stabilizes around 3kb/s (Fig. 17). The performance manager strives to maintain the desired throughput (8kb/s) for the high-priority during the remaining part of the test, with an average loss ratio below the threshold.

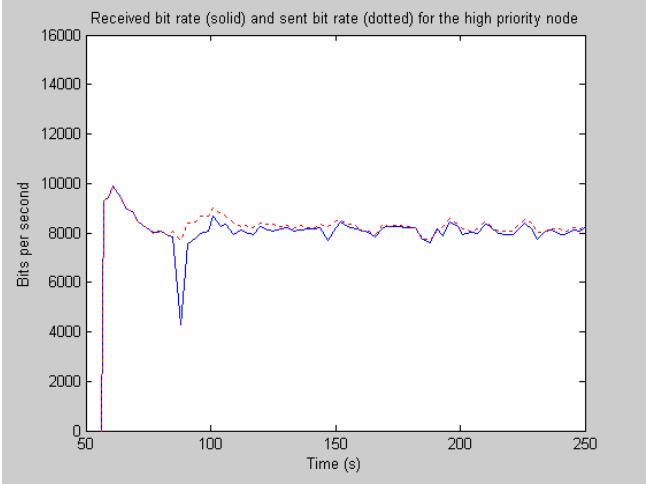

Figure 16. Throughput for the high-priority node.

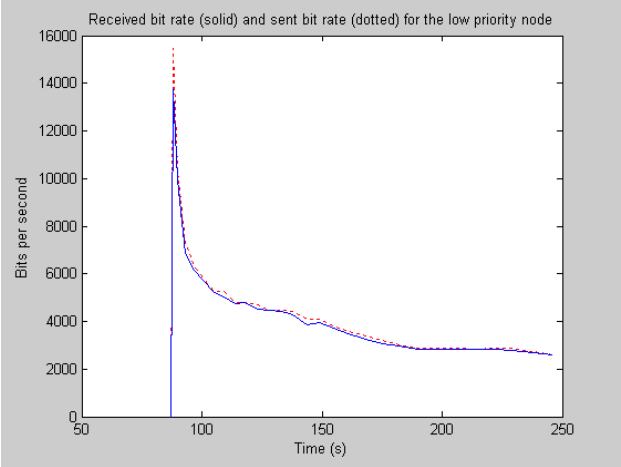

Figure 17. Throughput for the low-priority node.

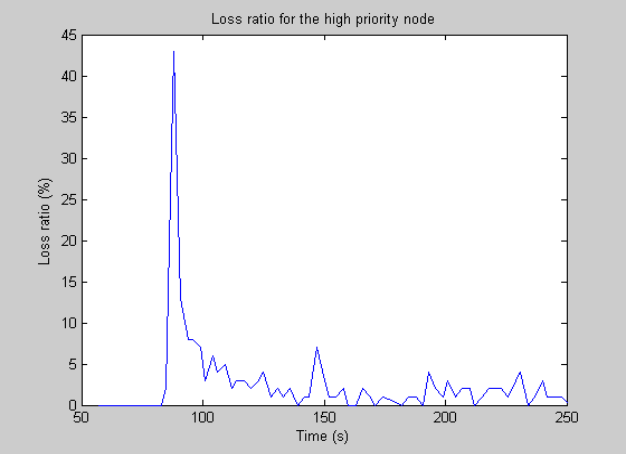

Figure 18. Loss ratio for the high-priority node.

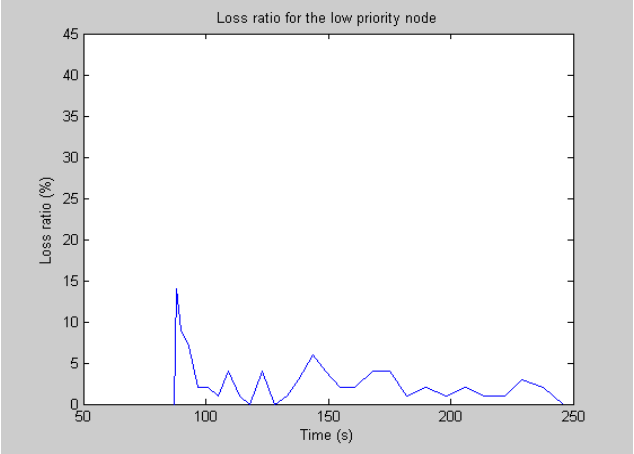

Figure 19. Loss ratio for the low-priority node.

2) Control without priority

In this case priority is not used. Both sensor nodes are controlled independently by the performance manager under the condition that the loss ratio is below a threshold. If the loss ratio exceeds the threshold, the sensor node will be instructed to decrease the sending rate (increase the packet interval by Δt milliseconds). No expected throughput is specified. Both sensor nodes start sending at 18kb/s as seen in Fig. 20 and Fig. 21. The high loss ratio for both nodes means that the performance manager will order both of them to slow down until the losses fall below the threshold. It can also be observed that the sensor node sometimes maintains the sending rate, even though the loss ratio is significantly higher than the threshold (Fig. 20 and Fig. 22). The explanation is that during heavy loss, monitoring packets will be lost as well, which delays the decision to decrease the packet frequency. After a while, the first node's throughput stabilizes around 3kb/s (Fig. 20) and around 3.5kb/s for the second node (Fig. 21). Since the control of the sensor nodes is independent of each other, the throughput will normally not be on the same level. One reason is different loss characteristics of the two channels; another may be different starting values. Each sensor node tries to find its maximum bit rate without exceeding the loss ratio threshold.

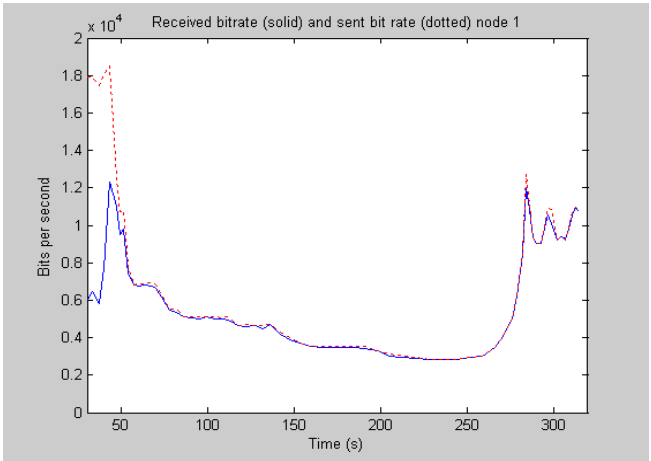

Figure 20. Throughput for node 1 (test case without priority).

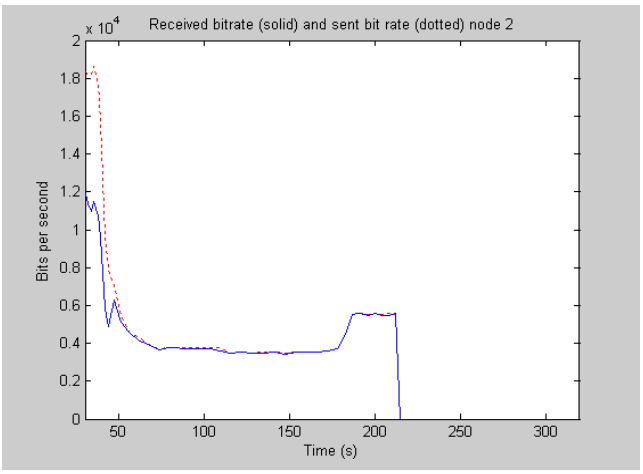

Figure 21. Throughput for node 2 (test case without priority).

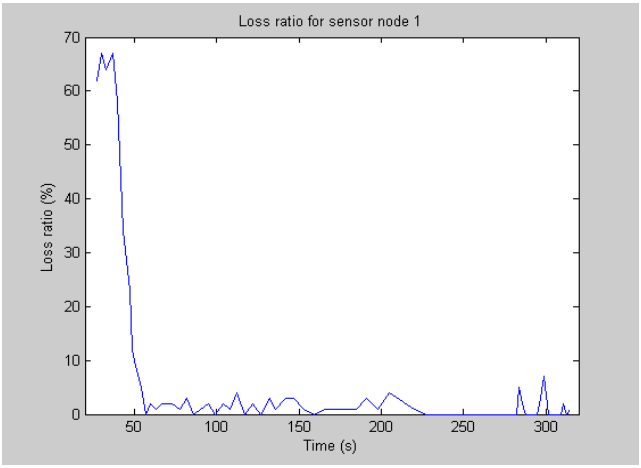

Figure 22. Loss ratio for node 1 (test case without priority).

At approximately t=180s, the manager has observed that the recent monitoring blocks are loss-free. The packet frequency is therefore increased for node 2 (Fig. 21). At t=210s, sensor node 2 stops transmitting (Fig. 21), which results in approximately zero packet loss for sensor node 1 (Fig. 22). The manager therefore tells the node to increase the packet frequency, up to around 10kb/s, where the loss threshold forces the node to slow down (Fig. 20).

3) Dynamic priority control

Fig. 23 and Fig. 24 show a combination of the previous two control algorithms. Both nodes start at a bit rate just below 15kb/s with 0.02 as the upper limit for the loss ratio. No node is given priority over the other. The throughput stabilizes between 4kb/s and 5kb/s. At t≈300 seconds, one of the nodes (Fig. 15) is dynamically assigned high priority, whereas the other node has to be satisfied with what is left. The reason might be that a higher sampling rate is needed for a sensor. The bit rate for the high-priority node rises to the required 8kb/s (Fig. 23) and the other sensor node backs off to around 2.5kb/s (Fig. 24). The step response in Fig. 23 takes around 30s. This time period can be reduced either by allowing larger step sizes (Δt) or decreasing the interval between the monitoring packets).

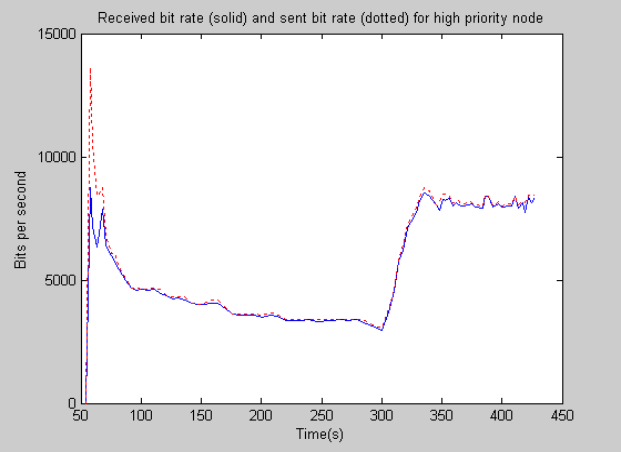

Figure 23. The sensor node is assigned high priority at t=300s and raises the bit rate to 8kb/s. The solid line represent received bit rate and the dotted line show sent bit rate.

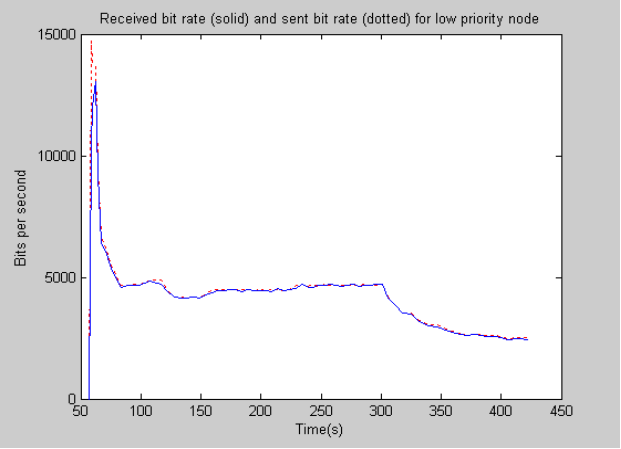

Figure 24. The sensor node is assigned low priority at t=300s and reduces the bit rate to 2.5kb/s. The solid line represent received bit rate and the dotted line show sent bit rate.

4) Multi-hop cases

The bit rate from a sensor node to a coordinator will to a large extent depend on the number of hops between the source and destination [4]. We have measured throughput between a sensor node and the receiving coordinator for zero, one and two intermediate nodes. The maximum received throughput for the equipment in our testbed using maximum packet length (payload 112 byte) was 50kb/s for one hop, 35kb/s for two hops and 20kb/s for three hops. This is of course a crucial limitation for demanding applications.

5) Results and discussion

Our analysis shows that it is feasible to use the measurement method, based on monitoring blocks, for performance monitoring as well as for feedback control of the performance of applications in wireless sensor networks. The results of the priority control algorithms are promising. The method has been implemented in a network with contention-based (CSMA/CA) access. It can of course also be used for the contention-based part of a super-frame in beacon mode in IEEE 802.15.4, where the contention-free part has guaranteed timeslots for the most demanding applications. One observation is that it is more straightforward to avoid packet loss in situations of buffer saturation by reducing the packet frequency, than to handle packet loss due to collisions and channel errors.

The monitoring and control method has three main parameters, that can be tuned for optimal results: the size of the monitoring block (B); the number of previous monitoring blocks (B_n, B_{n-1}, B_{n-2} etc) and their relative weight and, the step size (Δt) that controls the time interval between transmitted packets (or packet frequency). A more systematic study of these aspects related to control theory is for future work. To find out, in real-time, what capacity is available for a specified loss ratio, given that a second node transmits at a certain bit rate, is another example of application for the performance control method.

C. Admission Control and Performance Issues

In the following section we present some of the potential performance problems with contention-based access and the need for admission control, as well as continuing performance monitoring and control. Section V-C.1 illustrates the non-trivial performance problems associated with contention-based access (CSMA). Section V-C.2 shows how admission control works in real-time. The length of the test period is also discussed. In the third case (Section V-C.3), the implemented system is used as an off-line configuration tool to determine how changes of the traffic pattern influence the packet loss ratio. Finally, the alternative to allocate a new radio channel to a requesting sensor node is mentioned in Section V-C.4. The testbed consists of sensor nodes transmitting samples from ECGs, pulse-oximeters and accelerometers with sampling rates from 100Hz to 250Hz to a coordinator.

1) Performance problems in contention-based access

Contention-based access is a challenge for applications that require good and predictable performance. Fig. 25 illustrates what can happen when several sensors access a wireless channel. Three sensor nodes (A, B and C in Fig. 26) are connected to a coordinator sharing the same channel. The sensors are sampled during a second and the packets are sent back-to-back once a second. The bit rate is 9.6kbps for each sensor node. Fig. 25 shows the loss ratio during a measurement period for sensor node A. During the first part (0-70 seconds) only sensor node A is active. The loss ratio is almost zero. Between 70-140 seconds, sensor node B also accesses the channel. The average loss ratio experience by sensor node A is 0.03. During the remaining measurement period all three sensor nodes are transmitting on the same channel. The average loss ratio suddenly rises to 0.40.

For a loss-sensitive application, the performance is unacceptable after sensor node B, and especially after sensor node C, has joined the channel. The performance degradation may be avoided if the coordinator applies admission control and also maintains performance monitoring and control to protect the quality of service requirements for the existing nodes.

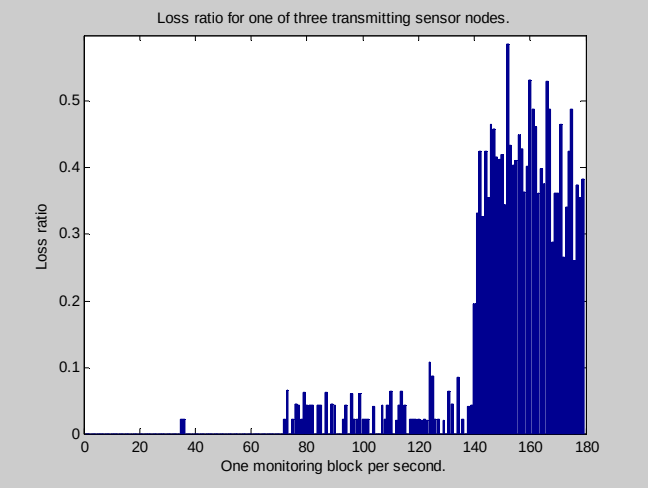

Figure 25. The loss ratio (y-axis) for sensor node A during the measurement period (x-axis in seconds). At approximately t=70s sensor B joins the channel. At t=140s a third node, sensor node C, accesses the channel.

2) Admission control

In this test case, the coordinator applies admission control when three sensor nodes with accelerometers, one by one, request to join the wireless network (Fig. 26). The sampling rate for the three-axis accelerometer is 200Hz per axis and the resulting average bit rate is 9.6kbps. The upper limit for packet loss for each node is set to 0.02 per monitoring block (the block length is around 1 second). The admission test period is 30 monitoring blocks (30 seconds). The measurement sequence is outlined in Fig. 27. Sensor node A requests admission and begins transmitting probe packets. The loss ratio during this admission test period is zero. Sensor node A's request is accepted and it starts transferring data. The loss ratio for the data traffic from sensor node A is almost zero before sensor node B requests to join the channel. Table IV summarizes the loss ratio for each sensor node during every test and data transfer period. Loss ratios exceeding the threshold (0.02) are indicated in bold text. It turns out that sensor node A and B are accepted, while sensor node C is rejected. For a sensor node to be rejected it is sufficient that the loss ratio for one of the sensor nodes, including the requesting node itself, exceeds the threshold.

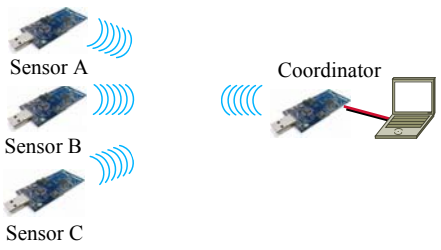

Figure 26. Three sensor nodes connected to the coordinator sharing the same channel.

The length of the test period is a trade-off between, on the one hand, to minimize the disturbance of existing traffic and reducing the response time for the admission verdict, and on the other hand, to receive sufficient performance data. The drawback of a predetermined fixed length of the test period is that ongoing traffic may suffer from severe performance deterioration. Fig. 28 shows the impact of test traffic on a sensor node during a 30 seconds test period. The average loss ratio is almost 0.05, with several peaks around 0.10, which is unacceptable performance deterioration for an already admitted node during a test period. To avoid this, we use an algorithm that calculates the cumulative moving average of the loss ratio for each incoming performance update i.e., for each monitoring packet. The test period is interrupted if the cumulative average exceeds a threshold. The cumulative moving average is defined as $CA_i=(L_1+L_2+L_3+...+L_i)/i$, where L_i is the loss ratio for monitoring block i. The algorithm is applied to the three test periods in Fig. 28 – Fig. 30. The cumulative averages for the first five blocks in Fig. 28 are $CA_1=0.059$, $CA_2=0.035$, $CA_3=0.032$, $CA_4=0.042$ and $CA_5=0.042$.

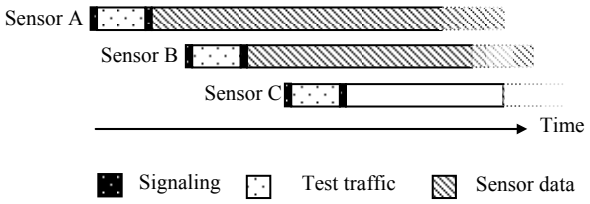

Figure 27. The measurement sequence for the nodes in Fig.26.

TABLE IV. LOSS RATIO FOR TEST PERIODS AND DATA TRANSFER PERIODS FOR SENSOR NODE A, B AND C.

	Sensor A	Sensor B	Sensor C
Test period sensor A	0.0000	--	--
Data transfer	0.0006	--	--
Test period sensor B	0.0012	0.0085	--
Data transfer	0.0019	0.0088	--
Test period sensor C	0.0046	**0.0470**	**0.0250**
Data transfer	0.0051	0.0083	--

If the rule for admittance is to allow maximum three consecutive updates of the loss ratio above the threshold (0.02), the test period will be interrupted after the third block. With an additional requirement that the loss ratio for a single block cannot exceed 0.05, this example means that the test period is interrupted after the first monitoring block.

A slightly different loss pattern is depicted in Fig. 29 (sensor node C's loss ratio during a test period). The cumulative average for the first seven blocks are $CA_1=0.0118$, $CA_2=0.0119$, $CA_3=0.0159$, $CA_4=0.0240$ and $CA_5=0.0216$, $CA_6=0.0201$ and $CA_7=0.0206$. In this case, the test period terminates after the 6^{th} monitoring block and the request is rejected.

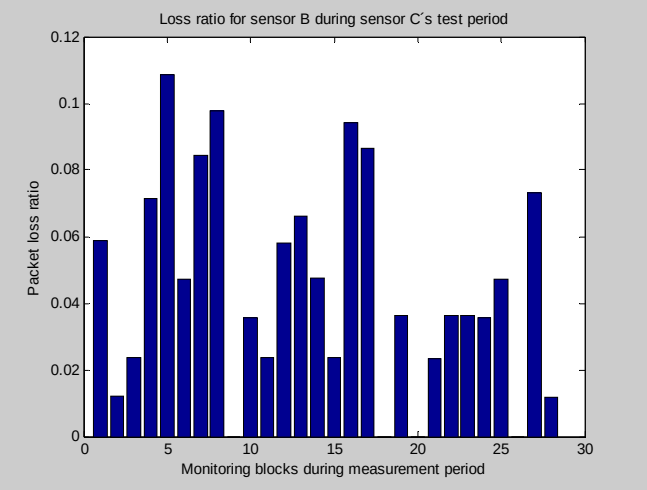

Figure 28. Loss ratio per monitoring block experienced by sensor node B during the third sensor's (sensor node C) test period. The average loss ratio is 0.047.

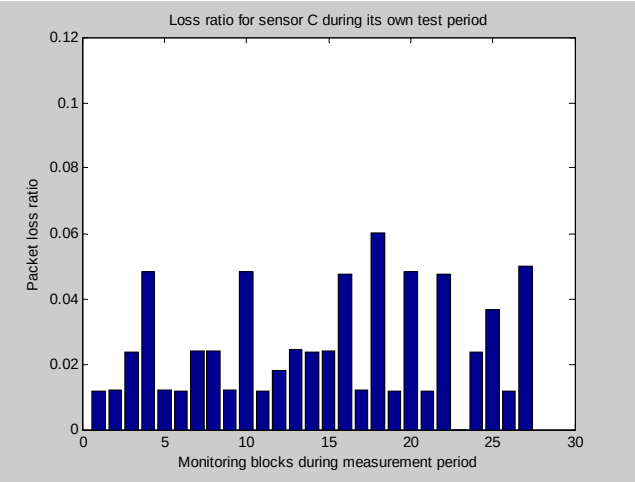

Figure 29. Loss ratio per monitoring block experienced by sensor node C during its own test period. The average loss ratio is 0.025.

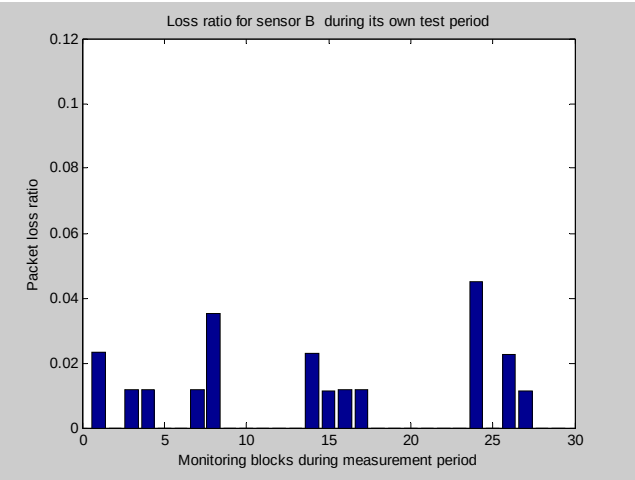

Figure 30. Loss ratio per monitoring block experienced by sensor node B during its own test period.

3) Traffic patterns and channel access

Packet loss in contention-based wireless networks is sensitive to the traffic pattern from the individual sources. Assume that two nodes collect samples and transmit the samples as a train of packets periodically once a second. If the nodes transmit the packet trains without overlap in time, the risk for losses due to collisions is low. However, the loss probability will increase if the packet trains happen to coincide. The dynamics of the traffic patterns in a network may from time to time lead to losses that exceed the accepted level after the admission test periods. The unpredictability of performance deterioration in wireless contention-based networks means that admission control must be combined with continuous traffic monitoring and control to be able to maintain the desired performance goals.

We have performed tests to study the impact of changes in traffic pattern on packet loss. Sensor node A collects and stores samples during a second. The samples are encapsulated in packets and transmitted back-to-back. The total time to transmit the packet train depends on the sampling rate, the sample size and the packet size. In this case, the sensor node sends a packet train of 43 packets with a packet size of 28 bytes, which corresponds to a throughput of 9.6kb/s. The total time to send the packet train was around 500ms. A second node, sensor node B, starts transmitting probe packets. It sends a train of packets once a second during the test period. The starting time for each train is shifted 50ms after ten seconds. This is repeated ten times, which means that the total time shift of the packet trains is around 500ms. The basic idea is to let the packet trains from sensor node B slide over the packet trains from sensor node A. Fig. 31 illustrates this convolution-like procedure.

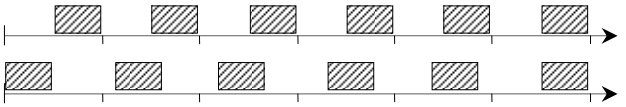

Figure 31. Sensor node A (the upper part) sends packet trains periodically every second. The starting times of the trains transmitted by sensor node B (lower part) are shifted in time so that they slide over the packet trains from sensor A.

Fig. 32 shows the loss ratio for sensor node B. After 10 monitoring blocks (10 seconds), the starting time is shifted 50ms. The average loss ratio for the first half of the measurement period is below 0.01. It rises to 0.10 for block 81-90 and 0.17 for block 91-100. The highest losses occur when the packet trains from the two sensors coincide in time. This convolution-like test might be inappropriate to use in an operating network but is useful for out-of-service configuration and dimensioning tests to estimate a worst case loss ratio. The traffic pattern for a channel e.g., the starting times of packet trains, is a stochastic process that may result in random losses from zero up to 0.25 in this case. Due to the unpredictability of contention-based wireless access continuous performance monitoring and control is needed to maintain the desired performance levels.

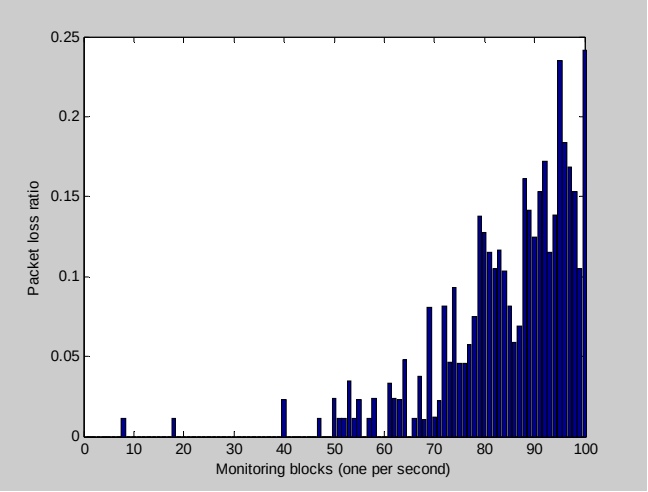

Figure 32. Loss ratio for sensor node B. The peak values occur when the packet trains from sensor node B coincide in time with the packet trains from sensor node A.

4) Redirecting to another channel

When a sensor node's request to join the network is rejected there are two alternatives. The node may back off for a while and try once again later. Alternatively, the coordinator may refer the sensor node to another radio channel. This feature has been successfully implemented and tested.

5) Results and discussion

The length of the test period is a trade-off between minimizing the disturbances on existing traffic, and receiving sufficient performance data for the admission verdict. The proposed algorithm uses a cumulative moving average of the loss ratio for the traffic from each sensor node to decide whether to reject an admission request and interrupt the test traffic, or to permit the sensor node to use the network. The test results show that admission control can improve the level, and predictability, of the performance of wireless sensor nodes. In addition, the method is also suitable for dimensioning, configuration and testing prior to operational mode. It can determine the number of sensor nodes that can share a wireless channel, for given performance requirements. A final conclusion is that continuous performance monitoring and control is needed to maintain the desired performance levels.

VI. CONCLUSIONS

Wireless sensor networks have today emerged as a feasible infrastructure for demanding applications e.g., in healthcare. This paper has addressed the non-trivial performance problems related to contention-based access to wireless channels. We have presented a measurement-based system for admission and performance control in wireless senor networks. The measurements are provided by a distributed light-weight performance meter. The test result shows that the implemented admission and performance control functions improve the quality, and predictability, of demanding services. The system can also be used as a tool for dimensioning and configuration of services in wireless sensor networks.

REFERENCES

[1] T. Lindh and I. Orhan, "Measurement-Based Admission Control in Wireless Sensor Networks", Sensorcomm, pp. 426–431, Venice/Mestre, July 2010.

[2] "Zigbee wireless sensor applications for health, wellness and fitness", Zigbee Alliance, March 2009.

[3] R. Carroll, R. Cnossen, M. Schnell, and D. Simons, "Continua: an Interoperable Personal Healthcare Ecosystem", IEEE Pervasive Computing, Vol. 6, No. 4, October-December 2007.

[4] A. Brandt (Zensys Inc) and G. Porcu (Telecom Italia), "Home Automation Routing Requirements in Low Power and Lossy Networks", Internet Draft, September 2009.

[5] I. Orhan, A. Gonga, and T. Lindh, "An End-to-End Performance Meter for Applications in Wireless Body Sensor Networks", BSN 2008, pp. 330–333, Hongkong, June 2008.

[6] T. Lindh and I. Orhan, "Performance Monitoring and Control in Contention-Based Wireless Sensor Networks", International Symposium on Wireless Communication Systems, pp. 507-511, Siena, Italy, September 2009.

[7] I. Más and G. Karlsson, "Probe-based admission control for differentiated-services internet", Computer Networks 51, pp.3902-3918, September 2007.

[8] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function", IEEE JSAC, Volume 18, No 3, pp 535 - 547 March 2000.

[9] Hai L. Vu, "Collision Probability in Saturated IEEE 802.11 Networks", Australian Telecommunication Networks and Applications Conference, pp. 21-25, Australia, December 2006.

[10] K. Duffy, D. Malone, and D.J. Leith, "Modeling the 802.11 Distributed Coordination Function in Non-saturated Conditions", Communications Letters, IEEE, Volume 9, Issue 8, pp. 715–717, August 2005.

[11] F. Österlind and A. Dunkels, "Approaching the Maximum 802.15.4 Multi-hop Throughput", HotEmnets, Virginia, June 2008.

[12] D. Cavalcanti et al, " Performance Analysis of 802.15.4 and 802.11e for Body Sensor Network Applications", BSN, Aachen, March 2007.

[13] N. Golmie et al: "Performance analysis of low rate wireless technologies for medical applications" Computer Communications, Volume 28, Issue 10, pp 1266-1275, June 2005.

[14] C.Y. Wan, S.B. Eisenman, and A.T. Campbell, "CODA: congestion detection and avoidance in sensor networks", SenSys, 1st conference on embedded networked sensor systems, pp 266-279, Los Angeles, November 2003.

[15] J. Kim, S. Kim, S, Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs", INFOCOM, pp 1-11, Barcelona, April 2006.

[16] P. Ramanathan, D. Moore, and C. Dovrolis "What Do Packet Dispersion Techniques Measure", In Proceedings of IEEE INFOCOM, 2001, pp. 905-914, Anchorage, Alaska, USA, April 2001.

[17] M. Jain and C. Dovrolis, "Pathload: a measurement tool for end-to-end available bandwidth", Passive and Active Measurements Workshop, pp 14-25, Fort Collins, USA, March 2002.

[18] S. Ekelin, M. Nilsson, E. Hartikainen, A. Johnsson, J-E Mångs, B. Melander, and M. Björkman, "Real-Time Measurement of End-to-End Available Bandwidth using Kalman Filtering", IEEE NOMS, pp 73-84, Vancouver, Canada, April 2006.

[19] T. Sun, L. Chen, G. Yang, M. Y. Sanadidi, and M. Gerla, "SenProbe: Path Capacity Estimation in Wireless Sensor Networks" SenMetrics 2005, San Diego, USA, July 2005

[20] D. Gupta, D. Wu, P. Mohapatra, and C-N. Chuah, "Experimental Comparison of Bandwidth Estimation Tools for Wireless Mesh Networks", IEEE INFOCOM Mini-Conference, pp 2891- 2895, April 2009.

[21] Y. Yang and R Kravets, "Contention-Aware Admission Control for Ad Hoc Networks" Mobile Computing, IEEE Transactions on Volume 4, Issue 4, pp 363 - 377, July-August, 2005.

[22] Ian D. Chakeres and Elizabeth M. Belding-Royer, "PAC: Perceptive Admission Control for Mobile Wireless Networks", International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine), pp 18-26, Dallas, USA, October 2004

[23] T. Lindh and N. Brownlee: "Integrating Active Methods and Flow Meters - an implementation using NeTraMet", Passive and Active Measurement workshop (PAM2003), San Diego, April 2003.

[24] M. Brenning, B. Olander, I. Orhan, J. Wennberg, and T. Lindh: "NeTraWeb: a Web-Based Traffic Flow Performance Meter", SNCNW2006, Luleå, Sweden, October 2006.

[25] "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, H. Schultzrinne et al., July 2003.

[26] D. Gay, P. Lewis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems", Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation, pp 1-11, San Diego, USA, June 2003.

[27] IEEE Standard 802.15.4 - 2006.

[28] Tmote Sky – IEEE 802.15.4 compliant sensor module from Sentilla (previously Moteiv).

[29] STB80 – Multi-Modality Sensor Board for TelosB Mote, www.easysen.com, May 2011.

Spectrum selection Through Resource Management in Cognitive Environment

Yenumula B. Reddy
Grambling State University, Grambling, LA 71245, USA
ybreddy@gram.edu

***Abstract*—Spectrum is a scarce resource. The cognitive radio environments utilize the spectrum efficiently through the dynamic spectrum access approach. Game theory, genetic algorithms, neural network, marketing, and economic models became the catalysts to boost the dynamic spectrum access for efficient utilization of the spectrum. Combinations of these models may sometimes help better in the allocation of unused spectrum (spectrum holes) for cognitive users (secondary users). The paper introduced the electronic commerce model to identify a quality channel preferred by the secondary user. The rating of the spectrum (channel) was modeled with a combination of Sporas formula and fuzzy reputation model proposed by Carbo, Molina and Davila. The proposed model helps to select the quality channel to cognitive user. Further, the cooperative game theory was introduced to gain the better channel by the cognitive user. The channel selection also tested with automated collaborative filtering and case-based reason applications.**

Keywords-dynamic spectrum access; spectrum holes; cooperative games, fuzzy rating, cognitive user.

1. INTRODUCTION

The existing static allocation of the spectrum is a major hurdle for its efficient utilization. Current research concentrates on efficient allocation of the allotted spectrum without disturbing the licensed users [1-3, 36]. For efficient allocation of the unused spectrum, one must create algorithms to track the unused spectrum at any given time, location, and allocate to the needed users. The design requires appropriate detectors at the physical layer to sense and access the strategies at the MAC layer. If the traffic on the communication space is random, the fixed channel allocation method is inefficient. The flexible sensing policy helps better in the allocation of the channels to the most qualified user (s) while avoiding other competing users.

For efficient use of the spectrum, one must detect the spectrum holes. The main function of cognitive user (CU) is sense, manage, mobile, and share the spectrum. It is difficult to search and detect the unused channels. The CU has to use a strategy to sense and manage the unused spectrum (channels). Game models help to detect the unused spectrum (available channels) and in assigning appropriate channel to the cognitive user.

Detecting spectrum holes (unused spectrum) without any errors (false alarms) and then efficiently allocating the unused spectrum is a critical issue. Once the unused spectrum is detected, the cognitive user (CU) decides to transmit on a selected high utility channel based on detection analysis (outcome). Trusting the detector, minimizing the interferences, and avoid the collisions are part of the resource (spectrum) allocation problems. If the probability of detection accuracy is extremely high, then the channel will be allocated using the channel allocation policies. The channel allocation policy must avoid the collisions and interferences.

The secondary users (cognitive users) adopt several methodologies to identify spectrum holes, learn from current communication environment, and exploit the opportunities to grab the spectrum without disturbing the primary user. This means that the cognitive users can create flexible access to the spectrum. They can partition the spectrum into a large number of orthogonal channels and complete the transmission simultaneously with a flexible set of channels. Such partition can be designed using a distributed approach for dynamic spectrum access. The approach may use graph models, game theory models, or any other Artificial Intelligence related models that quickly adapt to the varying traffic demands. These models determine the number of channels to use and maximize user throughput. These strategies follow the local maxima and continue until they achieve global maxima.

To use the spectrum holes efficiently, we must create the spectrum sharing scenario where multiple secondary users (SUs) coexist in the same area and communicate with the same portion of the spectrum. The game models were tested in a share market related problems and currently introduced in wireless communications [4-8]. The spectrum sharing problem can be solved with the cooperation of economic models and the selfish motivation of game models. In cooperative and selfish environments, the CU (SU) must consider the presence of other SUs as well as primary users (PUs). In a cooperative environment, it is required to consider explicit communication between CUs [9] only. In the selfish environment, the SUs compete for the resources using machine learning models or game models [1, 10, 11, 36].

In recent years, the FCC has been promoting the technologies for the efficient utilization of the spectrum. The FCC's interest helped industries to introduce devices with various capabilities including frequency agility, adaptive modulation, transmit power control, and localization. Furthermore, the NeXT generation program of the Defense Advanced Research Program Agency

(DARPA) concentrated on technology based on using cognitive radios (CR) for efficient utilization of the spectrum. Therefore, dynamic spectrum access (DSA) became a promising approach for efficient utilization of the spectrum [12].

Dynamic spectrum access (DSA) is the sharing of existing spectrum by unlicensed users (secondary or cognitive users) with licensed users (primary users) without interference to the licensed users. Sharing of spectrum requires following certain rules and policies. Spectrum sharing rules and protocols that allow the bandwidth to share are discussed in [14]. The important point is the amount of information that the secondary systems or cognitive systems need to know to use the unused spectrum. The information includes the current state of other cognitive users accessing same resource, unutilized and underutilized spectrum.

The key component for DSA is the sharing the spectrum with efficient allocation of the unused spectrum resources and the scheduling these resources among the SUs. Identification of the unused spectrum is the main part of the resource allocation. CRs play a major role in sensing unused spectrum intelligently making the appropriate decisions to gain access to unused spectrum. Game models help to select the unused spectrum intelligently and bring above the appropriate outcome for efficient allocation of resources. Since multiple users are involved in the process, security becomes an important factor. Therefore, CRs have additional work to deal with selfish and malicious users.

There are many models developed including efficient channel utilization, channel modeling, and allocation of resources for underlay techniques [6, 15, 22]. Recently, the research is diverted towards the design of efficient models for overlay and underlay techniques. The game models are the new introduction for spectrum sharing and utilization and produces encouraging results [30, 6, 15, 22]. The researchers introduced the role of games and game models for efficient use of spectrum. Non-cooperative and congestion game models are more suitable to the overlay and underlay spectrum usage.

Some of the game models presently trying to apply for overlay and underlay spectrum are zero-sum games, non zero-sum games, potential games, cooperative games, non-cooperative games, and congestion games. The behavior and stability of game models is measured through Nash equilibrium. The fundamental is that the players must reach Nash equilibrium to provide stable condition. Some of the models are discussed below:

- The zero-sum game is played between two players. The net result of two players that equals zero means that the gain of one player equates the loss of another player. Zero-sum game is not useful in the current problem because we have to utilize available spectrum efficiently. There is no loss involved.
- The non-zero-sum game model can be used in overlay spectrum utilization because if one or more of the players (cognitive users) cooperate then some or all of the players will be benefited. In non-zero-sum game, the cooperation of secondary users helps for efficient use of spectrum.
- The potential function in game model is a useful tool to analyze equilibrium properties of games. In potential game, the incentive of all players is mapped into a single function called potential function. In finite potential games [12, 13], the change in any player's utility exactly matches by a change in potential which concludes that the Nash equilibrium is a local maxima. If we treat the total incentive as utilizing unused spectrum, we may use the potential games to find the unused spectrum and make it available to the cognitive users.
- In cooperative games, the players stay close together for the overall benefit of all players. In non-cooperative games, each player makes independent decisions. Any cooperation in the game is self enforcing. The cooperative game may be useful in the current problem of efficient utilization of spectrum.
- Congestion games [15, 16] are a class of non-cooperative games where players share a common set of strategies. The utility of a player in congestion games depends on using a resource with the players that are using the same resource. That is the resulting payoff is a function of the number of active users (congestion). The authors used the congestion game model for better utilization of the unutilized spectrum by cognitive users.

In this research, marketing models for rating of a channel, case-based reasoning and automatic collaborative filtering concepts were used to allocate the preferred channel to the SU. The concept of the cooperative game approach boosts the case-based reasoning and automatic filtering models in maximizing channel allocation to a prioritized SU.

The Contribution:

The contribution includes the design of the problem using game models with opportunistic access to the spectrum by secondary users at any given time slot. The reward and penalty for user action are introduced depending upon the channel gain by the SU. The problem solution is dealt with game model using the collaborative effort of SUs. The channel rating was obtained using a new approach called Sporas, Carbo, Molina and Davila (SCMD) method. The channel gain by SUs was explained using Algorithm SCMD. The algorithm was discussed using simulations with random data developed in the MATLAB language.

The remaining part of the document is organized as follows: The related work and recent developments are

discussed in Section 2. The channel selection in Fuzzy environment was discussed in Section 3. The problem formulation for total reward was discussed in Section 4. Section 5 introduces the case-based reasoning and automatic collaborating filtering. Sections 6 and 7 discuss the cooperative game model and preference of a channel for secondary users. The algorithm and simulations using examples with sample data were discussed in Section 8. The Section 9 concludes the results and outlines the future research.

2. Related Work

The overview of the opportunistic spectrum access and taxonomy of dynamic spectrum access was provided by Zhao et al. [13]. Zhao et al. explained the confusion of the broad term CR as a synonym for dynamic spectrum access (DSA) and provides the clear distinction between the spectrum property rights, dynamic spectrum allocation, ultra wide band (spectrum underlay) and opportunistic spectrum access (spectrum overlay). Spectrum overlay (opportunistic spectrum access) and spectrum underlay techniques are also used for efficient allocation of the spectrum [31, 33]. Hidden Markov model to predict the primary user and efficient use of unused spectrum was discussed in [32]. Le and Hossain developed an algorithm for underlay and the quality of service (QoS) in code division multiple accesses (CDMA) with minimal interference was discussed in [34].

Efficient allocation of unused spectrum through auction, bidding, and rating was discussed in [1, 8, 19, 31, 32. The research contributions include the economic, game theory, stochastic, case based reasoning, Markov, and hybrid models for efficient allocation of unused spectrum. Auction-based dynamic spectrum allocation and lease the spectrum to CUs was discussed in [8]. The congestion game model for maximizing the spectrum utilization by secondary users with minimal interference to primary users was discussed in [31]. The Hidden Markov Models using Baum-Welch procedure to predict the future sequences infrequency and use them in computing the channel availability was discussed in [32]. The spectrum bidding behaviors and pricing models that maximize the revenue and better utilization of the spectrum was discussed in [19]. The market-based overlay model discussed in [7] imposes the spectrum mask to generate the better spectrum opportunities to secondary users. Furthermore, for efficient usage of unused spectrum, the primary user uses the economics competition for purchase of power allocation on each channel. This economic model uses the market equilibrium while controlling the interferences. The real-time spectrum auctions discussed by Gandhi et al. [19] include spectrum bidding behaviors and pricing models. The model discusses the spectrum demands and how to maximize the revenue for efficient utilization of the spectrum.

The recent developments for efficient utilization of spectrum holes include the role of cognitive radio using economic, game, stochastic, and Markov decision process models [14-18]. The auction-based dynamic spectrum allocation to lease the spectrum by secondary users was discussed by Wu [8]. In their paper, they proposed a mechanism called "multi-winner spectrum auction with a collision resistant pricing strategy" to allocate the spectrum optimally. The greedy algorithm proposed by Wu helps to reduce the complexity in multi-band auctions.

The game models for spectrum sharing and controlled interference is studied in [9, 20 -22]. Nie et al. [20] formulated the channel allocation as a potential game and showed the improvement of the overall network performance. Ji et al. [21] discussed the dynamic spectrum sharing through the game theory approach and discussed the analysis of the networks, the user's behavior, and the optimal analysis of the spectrum unused allocation. Halldorsson et al. [22] viewed channel assignment as a game and provided the price anarchy depending upon the assumptions of the underlying network. Finally, Liu et al. [9] used the congestion game model for spectrum sharing and base station channel adaptation.

In the current research, we introduced the opportunistic channel access by secondary users represented by a tuple consisting of a number of users contesting, the resources available, and the strategies used by the users that create an objective function (utility function). Whenever a user tries to gain an action to a channel, three cases arise. The user request is often accepted, denied, and the user has no action. The award or penalty depends after acceptance of the user request.

We further introduced the cooperative way of gaining a channel using SCMD model. These two models help to gain the channel access with the help of user's previous experience which includes channel weight and channel rating. The preference is calculated using nearest neighbor algorithm and/or mean square difference formula. These two formulas were used with users' cooperation activity. The cooperative activity includes their channels rating and user preferences. Using these concepts, the SU demands will be fulfilled.

3. Channel Selection in Fuzzy environment

Efficient allocation of the spectrum requires the coordination among the cognitive users (CU). The cognitive users prefer the high-rated spectrum. The high-rating depends upon the spectrum in demand. The high reputation of the spectrum is the measurement of demand. The reputation of the spectrum is the continuous rating by users. Sporas formula can be used for updating the reputation of the spectrum and CMD (Carbo, Molina and Davila) formula for updating fuzzy ratting. The rating of the spectrum with loosely connected agents (cognitive users) is given in the following formula [3].

$$R_i = R_{i-1} + \frac{1}{\theta}\phi(R_{i-1})(C_i - R_{i-1}) \qquad (1)$$

$$\phi(R_{i-1}) = 1 - \frac{1}{1 + e^{-(R_{i-1} - D)/\sigma}} \quad (2)$$

where:

θ - effective number of ratings taken into account ($\theta > 1$). The change in rating should not be very large.

ϕ - helps to slow down the incremental change

C_i - represents the rating given by the node i

D - range or maximum reputation value

σ - the acceleration factor to keep the ϕ above certain value (> threshold).

If the rating of the channel falls below the required threshold value ($(C_i - R_{i-1}) < 0$) then cognitive users will not request that channel. That is, the current channel cannot provide Quality of Service (QoS). The Sporas formula was used to rate the channel and verify the current rating. Initially $C_i - R_{i-1}$ was selected as positive and made C_i as random and > 0.9. The Figure 1 show the channel rating increases initially and falls down later, since the value of $C_i - R_{i-1}$ becomes negative at later time. We assumed the following arbitrary values for simulations to calculate the channel ratings.

θ=45; Ci > 0.9 (taken as random);

D=0.95; σ=0.3

R_{i-1}=0.95

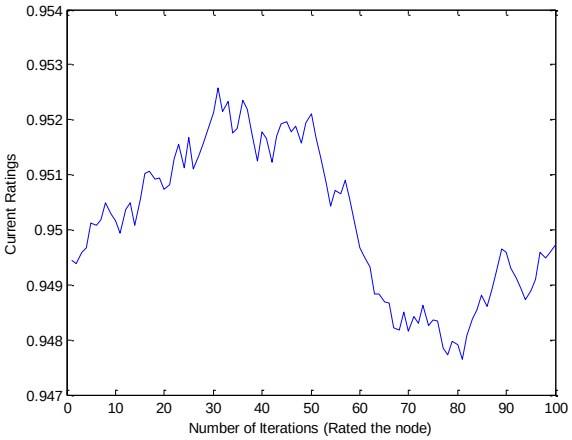

Figure 1: Channel Ratings using Sporas formula

Sporas formula calculates the ratings of a channel using the previous ratings. We have taken θ value bigger to observe the change of ratings of channel slowly.

The channel ratings are provided by the user. The ratings may be vague, uncertain, and incomplete. Representing such information in nature is fuzzy. The ratings of the channel are continuously changing through user supply information. The recent ratings (reputation of the channel) are modified (increase or decrease) using the user supplied approximate ratings. Therefore, current ratings depend upon the previous ratings.

Let R_{i-1} be the rating of the channel at $(i-1)^{th}$ instant and Z_i be the rating currently provided by the user after completion of the channel usage at instant i. The new reputation of the channel depends upon these values Z_i and R_{i-1}. The previous value will be used depends upon the learning factor ζ. Therefore, the reputation of the channel can be computed using the CMD formula [4, 5] as below.

$$R_i = \frac{R_{i-1}.\zeta + Z_i.(2 - \zeta)}{2} \quad (3)$$

If the learning factor $\zeta = 0$ means that the current rate is same as previous rate. If the learning rate $\zeta = 1$ means that the channel rate is constant. The learning rate will be updated each time by a factor δ using the Sporas formula. The update of learning factor is given as:

$$\zeta = \frac{\zeta + \delta}{2} \quad (4)$$

The value for δ is provided through Sporas formula from equation (1). The channel rate will be increased or decreased depending upon the current value ζ. Figure 2 shows the comparison of the channel ratings using Sporas formula, the fuzzy reputations using CMD formula, and CMD formula with learning updates using Sporas formula. The use of Sporas formula or CMD formula will not provide the stable ratings. Further, if updating the learning factor with Sporas formula, the channel rate will become stable quickly. The Figure 2 concludes that the best channel rate will be obtained with the combination of the recommendation process of CMD and learning updates with Sporas formula. The Figure 2 concludes that the Sporas formula or CMD formula will not provide the stable ratings of the channel. The combination of these

two models (hybrid model) will provide acceptable recommendation. The cognitive user will have a better choice if the user gets stable rating rather than vague ratings.

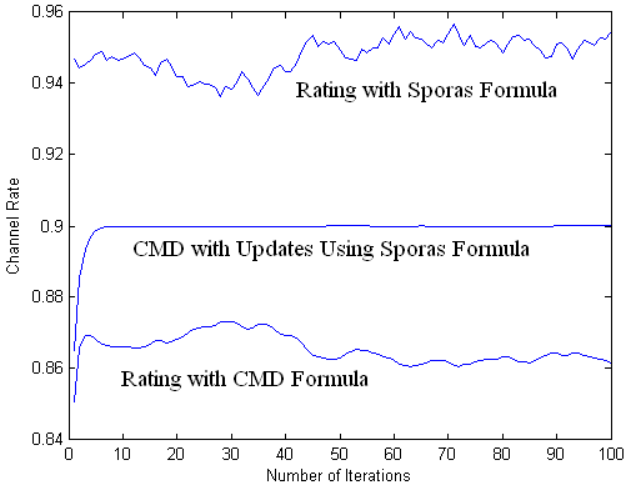

Figure 2: Comparison of Channel ratings (Sporas, CMD, and CMD& Sporas

4. PROBLEM

Let P denote the primary user (PU) and S denote the secondary users (SU). There are N primary users and M secondary users. Each primary user P_i $(i=1...N)$ has bandwidth $B_i (i=1,...,N)$ and secondary user S_i $(i=1...M)$ competes along with other secondary users $S_{j,j\neq i}$ in the network to gain access to unused spectrum by PUs. The SU or cognitive users (CU) compete for the spectrum during sleeping time (unused time) of PU. The available spectrum slots depend upon the geographical locations. Each SU competes for spectrum and sense it at the beginning of each time slot.

The availability of j^{th} primary channel $(j=1...N)$ for i^{th} secondary user $(i=1...M)$ depends upon the probability of the channel availability and the opportunistic access by the i^{th} SU. Suppose, there are κ primary channels available to M secondary users competing at any given time slot t, then each SU has chance to obtain the channel at time slot t is κ / M. Therefore, the opportunity for the i^{th} SU to access the κ available primary channels at any time slot t is

$$S_i^k(t) = \kappa / \mathrm{M} = K, \text{ for } i \in M$$

The total opportunities for all secondary users at any time slot for κ available primary channels is

$$\sum_{i=1}^{M} S_i^k(t) = \sum_{i=1}^{M} K_i(t) \text{ and } t \in T \text{ the time unit.}$$

The opportunistic channel access is represented by the tuple (S, K, Π, U), where S is the set of SUs competing for resources K $(k_i \in \mathrm{K})$ which is a set of channels available to SU at any given time slot t, and each SU uses the strategy $\pi \in \Pi$ that generates the utility $u \in U$. For each action of the secondary user S_i, the user senses the spectrum with a strategy π_i. The sensing action may get success/fail with reward R_i which is 1 or -1. The channel access tuple can be represented as a game, where each SU in the game uses a strategy to compete for the resource called spectrum that costs for gaining the specific utility as the object function. During the process the SU may gain or lose the opportunity and generate the reward or penalty. Therefore, the game model G is represented as

$$G = \{S, \mathrm{K}, \Pi, U\} \tag{5}$$

The total benefits are the sum of all gains minus the sum of all penalties. Therefore, the total reward $\Re$ is given by

$$\Re = \Lambda \sum_{i=1}^{M} \sum_{t=1}^{T} R_i^t \tag{6}$$

and Λ is a constant $0 \leq \Lambda \leq 1$.

$$R_i^t = \begin{cases} 1 & \text{if user } i \text{ gains resource with strategy } \pi_i \text{ at time slot } t \\ 0 & \text{for no action} \\ -1 & \text{if user } i \text{ fails to obtain the resource at time slot } t \end{cases} \tag{7}$$

The sensing policy of the cognitive user decides the action to be taken at a given time slot. Once the channel is sensed as free then the allocation policy decides which secondary user has priority to gain access. The average reward per time slot (T) is calculated as $\Re / T$ and will be used for throughput criteria.

The channel gained by the SU depends upon the current state and policy of the assignment that maximizes the total reward. Therefore, the channel assignment to the SU depends upon probability of availability of a channel and the probability of getting a channel. The probability of availability of a channel to a SU at any time slot t is

κ / M and the probability of assignment of the channel will be obtained by using the SCMD [23 - 25].

5. Case-Based Reasoning and Automatic Collaborative Filtering

In the selection of a product, CBR is used when experts find it hard to articulate their thought processes because the knowledge acquisition for these cases would be extremely difficult to such domains and is likely to produce incomplete or inaccurate results. CBR systems allow the case-base to establish characterized cases incrementally. Therefore, the case of SU is established incrementally for priority policy. A widely used formula for CBR in identifying and recommending similar channels (spectrum) is the nearest neighbor retrieval, which is based on weighted Euclidean distance [35].

The nearest neighbor algorithm deals with the similarity between the priority of stored cases (channels) and newly available cases (channel). The outcome depends upon the matching of the weighted sum of features. The toughest problem is to determine the weights of the features of the resource (spectrum). The limitation of this approach depends upon the converge to the correct solution and retrieval times. A typical algorithm for calculating the nearest neighbor (matching) is the one used by Cognitive Systems Remind software reported in Kolodner [24]. The nearest neighbor algorithm with weight of a i^{th} feature (w_i), similarity function sim, input case of i^{th} feature f_i^I , and retrieval case of i^{th} feature f_i^R is given by (Φ) [24]:

$$\Phi = \frac{\sum_{i=1}^{n} w_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} w_i} \qquad (8)$$

Equation (8) calculates the nearest neighbor of the best channel from the available channels of SU. If the difference between the i^{th} and j^{th} feature is negligibly smaller than the two features are closely matched. Equation (8) calculates the highly rated and close matching interests of the i^{th} user channel. The selection will be compared with ACF. The final selection will be the combined result of ACF and CBR.

Automated collaborated filtering (ACF) is a recommendation of a product based on word of mouth. In ACF, if user A's ratings of a channel (or channels) matches with another user B's ratings then it is possible to predict the ratings of a new channel for A, if B's rating for that channel is available. In other words, let us assume that if users X, Y, and Z have common interest in the channels C1, C2, and C3, then if X, Y did high rate of channel C4, then we can recommend the C4 for Z. That is, we can predict that user Z bids high for that channel C4, since C4 is close interest of Z. The approximate bid of a k^{th} bidder can be calculated by storing the bids of current bidders on the spectrum.

The ACF uses the mean squared difference formula [27] with two users. Let U and J be two SUs interested in a channel. Let U_f and J_f be the ratings of U and J on a feature f of the channel. Let χ be the set of features of the channel. Both U and J are rated and $f \in \chi$. The difference between two persons U and J in terms of their interests on a channel is given by [27]:

$$\Delta = \delta_{U,J} = \frac{1}{|\chi|} \sum_{f \in S} (U_f - J_f)^2 \qquad (9)$$

The ACF recommendations are two types, invasive and noninvasive based on the user preferences [1, 28, 29]. An invasive approach requires explicit user feedback, where the preferences can vary between 0 and 1. In the noninvasive approach, the preferences are interactive and Boolean values. In the noninvasive rating, zero (0) means the user does not rate the item and one (1) means rated. Therefore in noninvasive cases, it requires more data for any decision. In ACF systems, all user recommendations are taken into account, even though they are entered at different times. More user recommendations provide good strength for the ACF recommendation system and the new recommendations solely depends upon the data.

6. Cooperative Game Model

The secondary users rate the spectrum after the use. The rating will be updated (on a channel or spectrum) using SCMD. At a later time, the ratings will help the CUs to select appropriate spectrum to match their requirements. In the same way, the users rate the spectrum after they use in cooperative game model.

In cooperative games, the competition between coalitions of players groups the players and enforces cooperative behavior. The players choose the strategies by a consensus decision making process. It is assumed that each player has more than one choice. The combination of choices may win/loose/draw with an assigned payoff. The players know the rules and select the higher payoff. The payoff will be calculated using equation (7) and the channel selection with equations (8) and (9). This concludes that the channel selection will be done using the CBR and ACF models and cooperative behavior of the players.

The assignment of the channel to SU will depend upon the characteristic function ν as defined below:

Definition 1: The tuple (M, ν) is a cooperative game only if ν is monotone. This concludes that the cost assignment is positive. That is, $\nu(S) \leq \nu(S')$ for all $S \subseteq S' \subseteq M$.

The channel preferences can be arranged monotonically using the definition 1, equation (4), and equation (5). The SU will then choose the best choice from the available channels. For example, let us consider the similar interest SUs into a sub group S' where $S' \subseteq S$. The similar interest SUs on a particular available channel needs the priority of these users. The best available channel that will be calculated using equation (4) and equation (5) then provide the difference between two users in terms of their interest. Therefore, it is easy to assign the closely matching channel to one of the SU.

In a cooperative game, an allocation is simply a overall value created and received by a particular user. For example if x_i for $i = 1,2,3...n$, a collection of values related to a channel, then the allocation is efficient if x_i is in $\nu(S)$. That is

$$\sum_{i=1}^{n} x_i = \nu(S).$$

This shows that each player (SU) must get as much as they need without interacting with other users. The creation of quantity $\nu(S)$ means the efficiency of its created values and gain of access to the appropriate channel by a SU.

Definition 2: The marginal contribution of a player is the amount of the overall value created and the value shrinks if that player has to leave the game.

Therefore, in a collaborative effort, marginal contribution members deduce something about the overall contribution to a particular game. This is a justification for a cooperative game related to users' contribution and leads towards the justification of channel selection.

7. PREFERENCE OF THE CHANNEL

Let us assume that there are four SUs, rated using eight channels that were used before. The weights are the user ratings and the preferences are the similarity function sim. The sim is based on availability and retrieval (preference). Let the preference selection rate vary as $0 \le sim \le 1$. Similarly, the user rating (weight) also varies $0 \le w_i \le 1$. The preference of a channel is calculated with randomly selected values as in Table I. The Table 1 uses the abbreviations as below:

U# = User number
C# = Channel Number
CR = Channel Rating
CRR = Channel Retrieval Rate
w = weight assigned to channel
sim = Average weight

TABLE I: SPECTRUM BIDDING WITH N CHANNELS AND K USERS

U #	C#	CR	CRR	sim	w
1	1	0.5	0.6	0.55	0.6
2	2	0.4	0.4	0.4	0.5
1	4	0.6	0.5	0.55	0.5
4	3	0.8	0.7	0.75	0.7
3	5	0.2	0.3	0.25	0.3
2	8	0.5	0.6	0.55	0.6
1	7	0.8	0.8	0.8	0.9
2	6	0.6	0.6	0.6	0.7

The Φ value in equation (4) is for the nearest neighbor. The calculations are provided in table I and the Φ value is given by:

Φ =0.6948

Therefore, the channels closest and greater than the Φ value will have a better choice of the selection by the SU. In the above table, the channels 3 and 7 have primary choice and channels 1, 4, 6, and 8 will get second choice for selection. Channel 5 has low priority.

In the case of ACF, we will consider the interest of two users on a particular channel and their ratings. In this case, we will look into the quality of service (QoS) and ratings on a particular channel by the two users. The interest factor will be calculated using the channel features called QoS and Ratings as shown in Table II.

TABLE II CHANNEL RATING BY TWO USERS FOR PREFERENCE

User#	Rating	QoS	χ
1	0.7	0.8	0.9
2	0.8	0.75	0.9

Using the difference formula in equation (5), and the data from Table II, the Δ value is calculated and is given by:

Δ = 0.0139

In the current case, since the difference between the interests of two users is close to 1%, the two users will bid for the channel. The highest bidder wins the channel or highest preference by the priority allocation will get the channel. The preference (priority) may be a hand-shaking situation, where a user is moving from one access point to another. Therefore, in the current channel allocation the cooperative games use the economic model for preference

and allocation of the channel. The characteristic function is based on CBR or ACF model.

8. SIMULATIONS

We will discuss the efficient utilization of spectrum by secondary users by gaining a preferred. The following algorithm CBR-ACF calculates and assigns the appropriate channel using priorities to SUs at any given time.

Algorithm CBR-ACF

a) Find available channels (generate randomly)
b) Use the nearest neighbor algorithm (CBR) and
   - find the channel (s) closest to user1 preference
   - find the channel (s) closest to user 2 preference
c) Using ACF formula find the user preferences for the same available channels between two users
d) Assign the channel using CBR and ACF data
e) Repeat the steps (a) to (d) for another user

The Algorithm CBR-ACF selects the channel closest to the user choice. If more than one user is interested in the same channel, then the system must select the preferred user.

The simulations were conducted using random data. The program was written in MATLAB. In the current case, we assumed 99 channels. The values for CR, CRR, and w were assigned using a random function. The sim values are calculated for each channel. Many simulations were conducted and recorded for key explanation of algorithm CBR-ACF.

Case 1

The following data was obtained using the Algorithm CBR-ACF and equations (4) and (5). Basing the simulations data, we concluded the current available channel will be allocated to the preferred user.

Φ =0.5079
Free channels: 51 8 6 63 62
Preferred channel: 51
Preferred CRR for this channel: 0.5136
ACF for this channel calculated for 3 user cases:

User1 ACF = 0.0241
User2 ACF =0.0378
User3 ACF =0.0295

According to this data user 1 will be assigned the channel 51.

Case 2

Φ =0.4769
Free channels: 71 50 47 6 68
Preferred channel: 6
Preferred CRR for this channel: 0.4218
ACF for this channel calculated for 3 user cases:

User1 ACF = 0.0381
User2 ACF =0.0298
User3 ACF =0.0542

According to this data user 2 will be assigned the channel 6.

For example, consider the case 2; the CBR assigns the available channel to user 1 with its priorities. The priorities are calculated using the closest matching of ratings and combined decision using CBR and ACF. The decision is the collaborative since the outcome was used users' priorities and recording their suggestions through weights, channel ratings, and priorities. We are working on the reward and penalty depending upon the assignment. We are also working on the bidding policy, when two users are closely matches for request.

9. CONCLUSION

Recent developments show that the there are different approaches for the efficient utilization of the spectrum. The approaches include genetic algorithms, neural networks, stochastic models, game models, and economic models. These models are used display the by dynamic spectrum access concept as a catalyst for efficient utilization of unutilized and underutilized spectrum. The channel-gain in an opportunistic and collaborative way is also a part of efficient utilization of the spectrum. In the current research, we introduced a new approach using case-based reasoning and automatic collaborative filtering with the cooperative game approach. The combination brings the cooperative game concept in the selection of a channel. The research was presented as a basic game model and channel gain by the SUs. The channel assignment to SUs was done by using the CBR and ACF models. These approaches were demonstrated using examples through a proposed Algorithm CBR-ACF.

The future work involves the identification of preferred SUs using reward and punishment model. Furthermore, automatic collection of spectrum holes and prioritizing the SUs using economic and biological models will save time in the fast allocation of channels to SUs. Automatic prioritizing also saves time and improves the quality of channel assignment (to a preferred and needed user).

Acknowledgment

The research work was supported by Air Force Research Laboratory/Clarkson Minority Leaders Program through contract No: FA8650-05-D-1912. The author wishes to express appreciation to Dr. Connie Walton, Grambling State University, for her continuous support.

References

[1] Reddy, Y. B., "Efficient Spectrum Allocation Using Case-Based Reasoning and Collaborative Filtering Approaches", SENSORCOMM 2010, July 18-25, 2010, pp. 375-380.

[2] Wang, W., and Liu, X., "List-coloring based channel allocation for open spectrum wireless networks," in Proc. of IEEE VTC, 2005, pp. 690-694.

[3] Sankaranarayanan, S., Papadimitratos, P., Mishra, A., and Hershey, S., "A Bandwidth Sharing Approach to Improve Licensed Spectrum Utilization," in Proc. of the first IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005, pp. 279-288.

[4] MacKenzie, A., and DaSilva, L., Game Theory for Wireless Engineers. Morgan & Claypool Publishers, ISBN: 1-59829-016-9, 2006.

[5] Blumrosen, L., and Dobzinski,S., "Welfare Maximization in Congestion Games", ACM Conference on Electronics commerce (EC '06), Ann Arbor, Michigan, June 2006, pp. 1224-1236.

[6] Milchtaich, I., "Congestion Games with Player-specific Payoff Functions", Games and Electronic Behavior, 13, 1996, pp. 111-124.

[7] Xie, Y., "Competitive Market Equilibrium for Overlay Cognitive Radio", http://www.stanford.edu/~yaoxie/ee359win09, Stanford University, 2009 (last access: May 17, 2011),

[8] Wu, Y., Wang,B., Ray Liu,K.J., and Charles Clancy,T., "Collusion-Resistant Multi-Winner Spectrum Auction for Cognitive Radio Networks" IEEE Proc. of GLOBECOM 2008, pp. 1-5.

[9] Liu, H., Krishnamachari, B., and Zhao, Q., "Cooperation and Learning in Multiuser Opportunistic Spectrum Access", CogNets Workshop, IEEE ICC 2008, Beijing, China, pp. 487-492.

[10] Reddy, Y. B., "Detecting Primary Signals for Efficient Utilization of Spectrum Using Q-Learning", ITNG 2008, April 7 - 9, 2008, pp. 360-365.

[11] Reddy, Y. B., Smith, H., and Terrell, M., "Congestion game models for Dynamic Spectrum sharing", ITNG 2010, April 12 - 15, 2010, pp. 897-902.

[12] Akyildiz, I. F., Lee, W., Vuran, M. C., and Mohanty, S., "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey", Computer Neworks, Volume 50, September 2006, pp. 2127-2159.

[13] Zhao, Q., and Sadler, B., "A Survey of Dynamic Spectrum Access: Signal Processing, Networking, and Regulatory Policy", IEEE Signal Processing Magazine, Vol. 24, 79, 2007, pp. 79-89.

[14] Etkin, R., Parekh, A., and Tse, D., "Spectrum Sharing for Unlicensed Bands", IEEE Jr. on selected areas in communications, vol. 25, no. 3, 2007, pp. 517-528.

[15] Liu, M., and Wu, Y., "Spectrum Sharing as Congestion Games", 46th Allerton Conf. Comm. Control and Computing, Monticello, IL, Sept. 2008, pp. 1146-1153.

[16] Reddy, Y. B., "Cross-layer based Approach for Efficient Resource Allocation in Wireless Cognitive Networks", Distributed Sensor Networks, DSN 2008, November 16 – 18, 2008, Orlando, Florida, USA, pp. 385-390

[17] Liu, K., Xiao, X., and Zhao, Q., " Opportunistic Spectrum Access in Self Similar Primary Traffic", EURASIP Journal on Advances in Signal Processing: Special Issue on Dynamic Spectrum Access for Wireless Networkin, 2009, pp. 1-6.

[18] Liu, H., and Krishnamachari, B., "Cooperation and Learning in Multiuser Opportunistic Spectrum Access", CogNets Workshop, IEEE ICC 2008, Beijing, China, pp. 487 – 492.

[19] Gandhi, S., Buragohain, C., Cao, L., Zheng, H., and Suri, S., "A General Framework for Wireless Spectrum Auctions", 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, DySPAN 2007, April 2007, pp. 22-33.

[20] Nie, N., and Comaniciu, C., "Adaptive Channel Allocation Spectrum Etiquette for Cognitive radio Networks", Mobile Networks and Appls., 11, 2006, pp. 269 - 278.

[21] Ji, Z., and Liu, K. J. R., "Dynamic Spectrum Sharing: A Game Theoretical Overview", IEEE Comm. Magazine, May 2007, pp. 88-94.

[22] Halldorsson, M.M., Halperm, J. Y., Li Li, and Mirrokni, V.S., "On Spectrum Sharing Games", Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing, 2004, pp. 107-114.

[23] Chen, H., Wu, H., Hu, J., and Gao, C., "Agent-based Trust Management Model for Wireless Sensor Networks", International Conference on Multimedia and Ubiquitous Engineering", 2008, pp. 150-154.

[24] Carbo, J., Molina, J.M., and Davila, J., "Trust Management through Fuzzy Reputation", International Journal of Cooperative Information Systems", Vol. 12, Issue 1, 2003, pp. 135-155.

[25] Carbo, J., Molina, J.M., and Davila, J., "Comparing Predictions of Sporas vs. a Fuzzy Reputation System", 3rd International Conference on Fuzzy Sets and Fuzzy Systems, 200, pp. 147-153..

[26] Kolodner, J. L., Case-Based Reasoning, Morgan Kaufmann, 1993.

[27] Cunningham, P., "Intelligent Support for E-commerce", Proceedings of First International Symposium on Intelligent Systems, 2002, pp. 210-214.

[28] Hays, C., Cunningham, P., and Smyth, B., "A Case-based Reasoning View of Automated Collaborative Filtering", Case-Based Reasoning Research and Development, Lecture Notes in Computer Science, 2001, Volume 2080/2001, pp. 234-248

[29] Sollenborn, M., and Funk, P., "Category-Based Filtering and User Stereotype Cases to Reduce the Latency Problem in Recommender Systems", 6th European Conference on Case Based Reasoning, Springer Lecture Notes, 2002, Advances in Case-Based Reasoning, Lecture Notes in Computer Science, 2002, Volume 2416/2002, pp. 285-290.

[30] .8.Zhu Ji and K. J. Ray Liu., "Dynamic Spectrum Sharing: A Game Theoretical Overview", IEEE Comm. Magazine, May 2007, pp. 88-94.

[31] Y. B. Reddy and Heather Smith., "Congestion Game Model for Efficient Utilization of Spectrum", SPIE, 2010, 7707 - 9 V. 1 (p.1 to 12)

[32] M. Sharma, A. Sahoo, and K. D. Nayak., "Channel Modeling Based on Interference Temperature in Underlay Cognitive Wireless Networks", ISWCS, 2008, pp. 224-228.

[33] V. D. Chakravarthy., "Evaluation of Overlay/Underlay Waveform via SD-SMSE Framework for Enhancing Spectrum Efficiency", Ph. D. Thesis, Wright State University, 2008.

[34] L. B. Le and E. Hossain., "Resource Allocation for Spectrum Underlay in Cognitive radio Networks", IEEE Transaction on Wireless Communications, Vol. 7, No. 2, 2008, pp. 5306-5315.

[35] Wettschereck, D., and Aha, D. W., "Weighting Features", Proc. Of the 1st International Conference on Case-Based Reasoning, Springer, New York, USA, 1995, pp. 347-358.

[36] Zheng, H., and Peng, C., "Collaboration and Fairness in Opportunistic Spectrum Access", in Proc. of IEEE Int. Conf. on Comm. (ICC), 2005, pp. 3132-3136.

.

A Practical Approach to Uncertainty Handling and Estimate Acquisition in Model-based Prediction of System Quality

Aida Omerovic*† and Ketil Stølen*†
*SINTEF ICT, P.O. Box 124, 0314 Oslo, Norway
†University of Oslo, Department of Informatics, P.O. Box 1080, 0316 Oslo, Norway
Email: {aida.omerovic,ketil.stolen}@sintef.no

***Abstract*—Our earlier research indicated the feasibility of applying the PREDIQT method for model-based prediction of impacts of architectural design changes on system quality. The PREDIQT method develops and makes use of so called prediction models, a central part of which are the "Dependency Views" (DVs) – weighted trees representing the relationships between architectural design and the quality characteristics of a target system. The values assigned to the DV parameters originate from domain expert judgements and measurements on the system. However fine grained, the DVs contain a certain degree of uncertainty due to lack and inaccuracy of empirical input. This paper proposes an approach to the representation, propagation and analysis of uncertainties in DVs. Such an approach is essential to facilitate model fitting (that is, adjustment of models during verification), identify the kinds of architectural design changes which can be handled by the prediction models, and indicate the value of added information. Based on a set of criteria, we argue analytically and empirically, that our uncertainty handling approach is comprehensible, sound, practically useful and better than any other approach we are aware of. Moreover, based on experiences from PREDIQT-based analyses through industrial case studies on real-life systems, we also provide guidelines for use of the approach in practice. The guidelines address the ways of obtaining empirical estimates as well as the means and measures for reducing uncertainty of the estimates.**

***Keywords*-uncertainty, system quality prediction, modeling, architectural design, change impact analysis, simulation.**

I. Introduction

An important aspect of quantitative prediction of system quality lies in the appropriate representation, propagation and interpretation of uncertainty. Our earlier work has addressed this issue by proposing an interval-based approach to uncertainty handling in model-based prediction of system quality [1]. This paper extends the interval-based approach to uncertainty handling with two major tightly related issues:

- uncertainty analysis, and
- practical guidelines for use of the interval-based approach, addressing both the uncertainty handling and the estimate acquisition.

We have developed and tried out the PREDIQT method [2], [3] for model-based prediction of impacts of architectural design changes on system quality characteristics and their trade-offs. Examples of quality characteristics include availability, scalability, security and reliability. Among the main artifacts of the PREDIQT method are the Dependency Views (DVs). The DVs currently rely on sharp parameter values which are based on empirical input. As such, the parameters assigned to the DVs are not very reliable, thus providing predictions of unspecified certainty.

Since the input to the DVs is based on both measurement-based data acquisition (measurements, logs, monitoring, historical data, or similar) and expert judgements, the representation of the uncertain input should be intuitive, as exact as possible and provide a well defined (complete and sound) inferring mechanism. In a real-life setting, finding the right balance between accuracy and practical feasibility is the main challenge when selecting the appropriate approach to uncertainty handling in prediction models. We propose an approach to deal with uncertainty which, as we will argue, is both formally sound and practically applicable in the PREDIQT context. Our approach is based on intervals with associated confidence level, and allows representation, propagation and analysis of all the parameters associated with uncertainty.

Input acquisition is in this context concerned with how the DV estimates and their uncertainty measures are obtained in practice. An overview of the practical means and measures for 1) acquiring the input and 2) achieving a specified minimum level of uncertainty, is clearly a prerequisite for applicability of the uncertainty handling approach. Therefore, we also provide guidelines for practical use of our solution, covering both the issues of estimate acquisition and uncertainty handling. The guidelines build on the experiences from the empirical evaluations of the PREDIQT method.

The paper is organized as follows: The challenge of uncertainty handling in the context of the PREDIQT method is characterized in Section II. We define the frame within which the approach should be applicable, by providing an overview of the PREDIQT method and in particular the DVs, introducing the notion of uncertainty, and outlining a set of success criteria. The interval-based approach to uncertainty handling is presented in Section III. Section IV argues for the usefulness and practical applicability of the approach by evaluating it with respect to the success criteria. An extensive

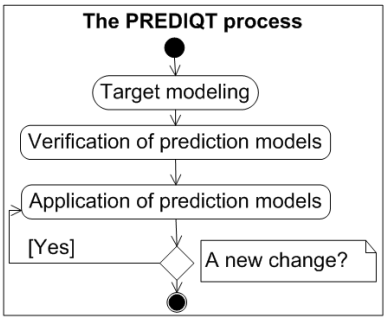

Figure 1. The overall PREDIQT process

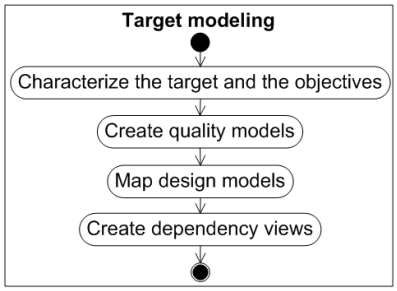

Figure 2. Target modeling phase

number of the candidate methods for uncertainty handling have been systematically reviewed prior to the proposal of our approach. Section V substantiates why our approach, given the criteria outlined in Section II, is preferred among the alternative ones. Practical guidelines for use of our solution, based on lessons learned from PREDIQT-based analyses on real-life systems, are provided in Section VI. The concluding remarks and the future work prospects are given in Section VII.

II. The challenge

Our earlier work indicates the feasibility of applying the PREDIQT method for model-based prediction of impacts of architectural design changes, on the different quality characteristics of a system. The PREDIQT method produces and applies a multi-layer model structure, called prediction models. The PREDIQT method is outlined in the next subsection. Uncertainty and the evaluation criteria for the uncertainty handling approach are thereafter presented in dedicated subsections.

A. Overview of the PREDIQT method

The PREDIQT method defines a process and a structure of prediction models. These two perspectives are presented in the following.

1) The process of the PREDIQT method: The process of the PREDIQT method consists of three overall phases as illustrated by Figure 1. Each of these phases is decomposed into sub-phases.

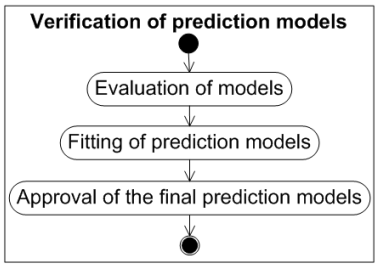

Figure 3. Verification of models – phase

The sub-phases within the "Target modeling" phase are depicted in Figure 2. Based on the initial input, the stakeholders involved deduce a high-level characterization of the target system, its scope and the objectives of the prediction analysis, by formulating the system boundaries, system context (including the usage profile), system lifetime and the extent (nature and rate) of design changes expected. Quality Model diagrams are created in the form of a tree, by decomposing total quality into the system specific quality characteristics, their respective sub-characteristics and indicators. The Quality Model diagrams represent a taxonomy with interpretations and formal definitions of system quality notions. The initially obtained Design Models are customized so that (1) only their relevant parts are selected for use in further analysis; and (2) a mapping within and across high-level design and low-level Design Models (if available), is made. The mapped models result in a class diagram, which includes the relevant elements and their relations only. A conceptual model (a tree-formed class diagram) in which classes represent elements from the underlying Design Models and Quality Models, relations represent the ownership, and the class attributes represent the dependencies or the properties, is created.

For each quality characteristic defined by the Quality Model, a quality characteristic specific DV is created via the instantiation of the conceptual model. A DV is basically a weighted dependency tree which models the relationships among quality characteristics and the design of the system. The instantiation of the conceptual model into a DV is performed by selecting the elements and relationships which are relevant to the quality characteristic being addressed by the DV. Each set of nodes having a common parent is supplemented with an additional node called "Other" for completeness purpose. The DV parameters are assigned by providing the estimates on the arcs and the leaf nodes, and propagating them according to a pre-defined inference algorithm.

The sub-phases within the "Verification of prediction models" phase are depicted in Figure 3. This phase aims to validate the prediction models, with respect to the structure and the individual parameters, before they are applied. A measurement plan with the necessary statistical power is

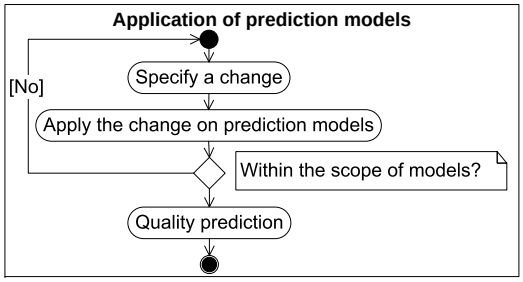

Figure 4. Application of models – phase

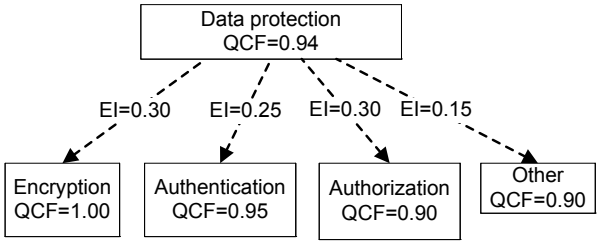

Figure 5. Excerpt of an example DV with fictitious values

developed, describing what should be evaluated, when and how. Both system-as-is and change effects should be covered by the measurement plan. Model fitting is conducted in order to adjust the DV structure and the parameters to the evaluation results. The objective of the "Approval of the final prediction models" sub-phase is to evaluate the prediction models as a whole and validate that they are complete, correct and mutually consistent after the fitting. If the deviation between the model and the new measurements is above the acceptable threshold after the fitting, the target modeling phase is re-initiated.

The sub-phases within the "Application of prediction models" phase are depicted in Figure 4. This phase involves applying the specified architectural design change on the prediction models and obtaining the predictions. The phase presupposes that the prediction models are approved. During this phase, a specified change is applied on the Design Models and the DVs, and its effects on the quality characteristics at the various abstraction levels are simulated on the respective DVs. The change specification should clearly state all deployment relevant facts necessary for applying the change. The "Apply the change on prediction models" sub-phase involves applying the specified architectural design change on the prediction models. When an architectural design change is applied on the Design Models, it is according to the definitions in the Quality Model, reflected to the relevant parts of the DVs. Thereafter, the DVs provide propagation paths and quantitative predictions of the new quality characteristic values, by propagating the change throughout the rest of each one of the modified DVs, based on the general DV propagation algorithm. We have earlier developed tool support [2] based on MS Excel [4] for simulation and sensitivity analysis of DVs.

The intended application of the prediction models does not include implementation of change on the target system, but only simulation of effects of the independent architectural design changes on quality of the target system (in its currently modelled state). Hence, maintenance of prediction models is beyond the scope of PREDIQT.

2) The prediction models: The PREDIQT method produces and applies a multi-layer model structure, called prediction models, which represent system relevant quality concepts (through "Quality Models") and architectural design (through "Design Models").

The Design Models represent the architectural design of the target system. The models include the parts and the detail level characterized (during the first sub-phase of the PREDIQT process) as a part of the objective of the analysis. Typically, Design Models include diagrams representing the process, the system structure, the dataflow and the rules for system use and operation. The Design Model diagrams are used to specify the target system and the changes whose effects on quality are to be predicted.

A Quality Model is a tree-like structure whose nodes (that is, quality notions at the different levels) are defined qualitatively and formally, with respect to the target system. The total quality of the system is decomposed into characteristics, sub-characteristics and quality indicators. Each of them is, by the Quality Model, defined in terms of a metric and an interpretation with respect to the target system. The definitions of the quality notions may for example be based on ISO 9126 product quality standard [5].

In addition, the prediction models comprise DVs, which are deduced from the Design Models and the Quality Models of the system under analysis. As explained above, the DVs model the dependencies of the architectural design with respect to the quality characteristic that the DV is dedicated to, in the form of multiple weighted and directed trees. The values and the dependencies modeled through the DVs are based on the quality characteristic definition provided by the Quality Model. A DV comprises two notions of parameters:

1) EI: Estimated degree of Impact between two nodes, and
2) QCF: estimated degree of Quality Characteristic Fulfillment.

Each arc pointing from the node being influenced is annotated by a quantitative value of EI, and each node is annotated by a quantitative value of QCF.

Figure 5 shows an excerpt of an example DV with fictitious values. In the case of the *Encryption* node of Figure 5, the QCF value expresses the goodness of encryption with respect to the quality characteristic in question, e.g., security. A QCF value on a DV expresses to what degree the node (representing system part, concern or similar) is realized so that it, within its own domain, fulfills the quality characteristic. The QCF value is based on the formal definition

of the quality characteristic (for the system under analysis), provided by the Quality Models. The EI value on an arc expresses the degree of impact of a child node (which the arc is directed to) on the parent node, or to what degree the parent node depends on the child node. The EI of an arc captures the impact of the child node on its parent node, with respect to the quality characteristic under consideration.

Input to the DV parameters may come in different forms (e.g., from domain expert judgements, experience factories, measurements, monitoring, logs, etc.), during the different phases of the PREDIQT method. Once the initial parameter values are assigned, the QCF value of each non-leaf node is recursively (starting from leaf nodes and moving upwards in the tree) propagated by multiplying the QCF and EI value for each immediate child and summing up these products for all the immediate children. This is referred to as the general DV propagation algorithm. For example, with respect to *Data protection* node in Figure 5 (denoting: DP: Data protection, E: Encryption, AT: Authentication, AAT: Authorization, and O:Other):

$$QCF_{(DP)} = QCF_{(E)} \cdot EI_{(DP \rightarrow E)} + QCF_{(AT)} \cdot EI_{(DP \rightarrow AT)} + QCF_{(AAT)} \cdot EI_{(DP \rightarrow AAT)} + QCF_{(O)} \cdot EI_{(DP \rightarrow O)} \quad (1)$$

The DV-based approach constrains the QCF of each node to range between 0 and 1, representing minimal and maximal characteristic fulfillment (within the domain of what is represented by the node), respectively. This constraint is ensured through the normalized definition of the quality characteristic metric. The sum of EIs, each between 0 (no impact) and 1 (maximum impact), assigned to the arcs pointing to the immediate children must be 1 (for model completeness purpose). Moreover, all nodes having a common parent have to be orthogonal (independent). The dependent nodes are placed at different levels when structuring the tree, thus ensuring that the needed relations are shown at the same time as the tree structure is preserved. The overall concerns are covered by the nodes denoted *Other*, which are included in each set of nodes having a common parent, thus making the DV complete.

The general DV propagation algorithm, exemplified by Eq. 1, is legitimate since each quality characteristic DV is complete, the EIs are normalized and the nodes having a common parent are orthogonal due to the structure. A DV is complete if each node which is decomposed, has children nodes which are independent and which together fully represent the relevant impacts on the parent node, with respect to the quality characteristic that the DV is dedicated to.

The rationale for the orthogonality is that the resulting DV structure is tree-formed and easy for the domain experts to relate to. This significantly simplifies the parameterization and limits the number of estimates required, since the number of interactions between the nodes is minimized. Although the orthogonality requirement puts additional demands on the DV structuring, it has been shown to represent a significant advantage during the estimation.

Figure 6 provides an overview of the prediction models, expressed as a UML [6] class diagram. A prediction model is decomposed into a Design Model, a Quality Model and a DV. A Quality Model is a set of tree-like structures. Each tree is dedicated to a target system-relevant quality characteristic. Each quality characteristic may be decomposed into quality sub-characteristics, which in turn may be decomposed into a set of quality indicators. As indicated by the relationship of type aggregation, specific sub-characteristics and indicators can appear in several Quality Model trees dedicated to the different quality characteristics. Each element of a Quality Model is assigned a quantitative normalized metric and an interpretation (qualitative meaning of the element), both specific for the target system. A Design Model represents the relevant aspects of the system architecture, such as for example process, dataflow, structure and rules. A DV is a weighted dependency tree dedicated to a specific quality characteristic defined through the Quality Model. As indicated by the attributes of the Class *Node*, the nodes of a DV are assigned a name and a QCF (that is, value of the degree of fulfillment of the quality characteristic, with respect to what is represented by the node). As indicated by the *Semantic* dependency relationship, semantics of both the structure and the weights of a DV are given by the definitions of the quality characteristics, as specified in the Quality Model. A DV node may be based on a Design Model element, as indicated by the *Based on* dependency relationship. As indicated by the self-reference on the Class *Node*, one node may be decomposed into children nodes. Directed arcs express dependency with respect to quality characteristic by relating each parent node to its immediate children nodes, thus forming a tree structure. Each arc in a DV is assigned an EI, which is a normalized value of degree of dependence of a parent node, on the immediate child node. The values on the nodes and the arcs are referred to as parameter estimates. We distinguish between prior (or initial) and inferred parameter estimates. The former ones are, in the form of empirical input, provided on leaf nodes and all arcs, while the latter ones are deduced using the DV propagation model for PREDIQT exemplified above. For further details on the PREDIQT method, see [2], [3], [7], [1].

B. Uncertainty

The empirical input is always associated with a degree of uncertainty. Uncertainty is generally categorized into two different types: aleatory (due to inherent randomness of the system or variability of the usage profile) and epistemic (due to lack of of knowledge or information about the system) [8]. The aleatory uncertainty is irreducible even by additional measurements. Aleatory uncertainty is typically represented by continuous probability distributions and forecasting is

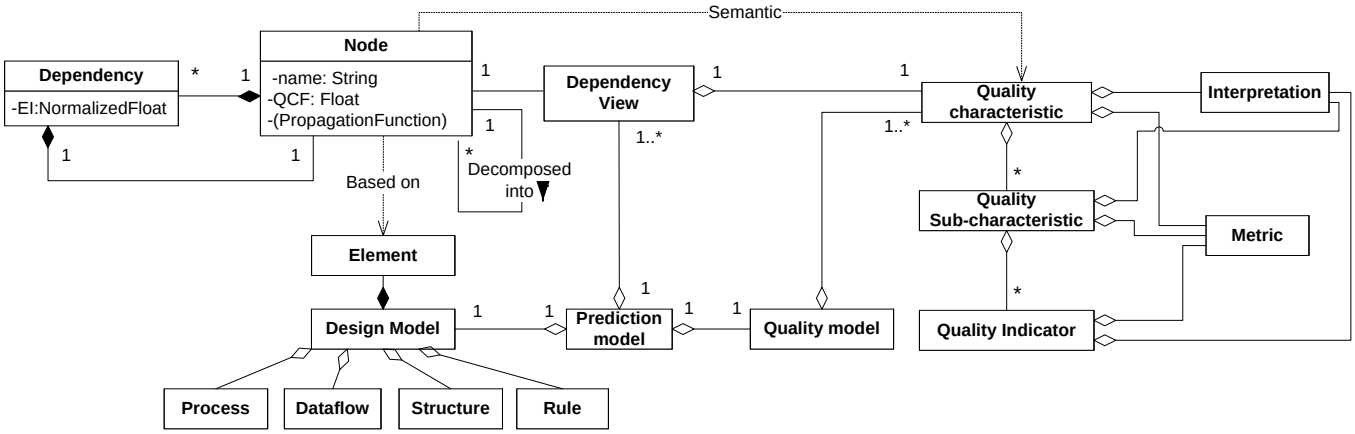

Figure 6. An overview of the elements of the prediction models, expressed as a UML class diagram

based on stochastic models.

Epistemic uncertainty, on the other hand, is reducible, non-stochastic and of discrete nature. The epistemic uncertainty is therefore best suited for possibilistic uncertainty representations. For a detailed classification of the types and sources of imperfect information, along with a survey of methods for representing and reasoning with the imperfect information, see [9]. For a systematic literature review of the approaches for uncertainty handling in weighted dependency trees, see [10].

Prediction models, as opposed to for example weather forecasting models, are characterized by rather discrete, sudden, non-stochastic and less frequent changes. The weather forecasting models are of stochastic and continuous nature and the aleatory uncertainty is the dominating one (due to uncontrollable variabilities of many simultaneous factors). In majority of the system quality prediction models, aleatory uncertainty is negligible in terms of magnitude and impact, while the epistemic one is crucial. It is therefore the epistemic uncertainty we focus on when dealing with the parameters on the DVs.

C. Success criteria

Since expert judgements are a central source of input during the development of the prediction models, and also partially during the model verification, it is crucial that the formal representation of uncertainty is comprehensible to those who have in-depth system knowledge, but not necessarily a profound insight into the formal representation. The representation form of uncertainty estimates should make them easy for domain experts to provide and interpret.

Simultaneously, each individual parameter estimate should express the associated uncertainty so that it is as exact as possible. That is, the parameter and uncertainty values provided should be as fine grained as possible to provide, but without restricting comprehensibility. Thus, the right granularity of the uncertainty representation at the level of each parameter is needed.

Moreover, the input representation should facilitate combining both expert judgement-based and measurement-based input at the level of each parameter in a DV.

The DV propagation algorithm has a number of associated prerequisites (e.g., completeness, independence of the nodes which have a common parent, and ranges that the EI and QCF values can be expressed within). Together, they restrict the inference and the structure of the DVs so that the DVs become sound and comprehensible. When the parameters with the uncertainty representation are propagated within and across the DVs, the inference must still be well-defined and sound.

When applied on real-life cases, the uncertainty handling approach should propagate to practically useful predictions, in the sense that the approach can be applied on realistic DVs with limited effort and give valuable output.

Statistical and sensitivity analyses are currently performed in the DVs, during the *Fitting of prediction models* subphase and the *Application of prediction models* phase (of the PREDIQT process), respectively. Therefore, the uncertainty handling approach should also allow deduction of the central tendency measures such as mode, median, arithmetic mean, geometric mean, and variance.

Given the overall objective and context, the main success criteria for the uncertainty handling approach can, in a prioritized order, be summarized into:

1) The representation form of each parameter estimate and its uncertainty should be comprehensible for the domain experts involved in the development and use of the prediction models.
2) The representation form of each parameter estimate and its uncertainty should be as exact as possible, in terms of expressing both the parameter estimate and the associated uncertainty.

3) The approach should facilitate combining both expert judgement-based and measurement-based input.
4) The approach should correctly propagate the estimates and their uncertainty.
5) The approach should provide practically useful results.
6) The approach should allow statistical analysis.

III. Our solution

This section presents an interval-based approach to representation and propagation of uncertainties on the DVs.

A. Uncertainty representation

All prior estimates (the terms "prior estimate" and "initial estimate" are used interchangeably, and regard the intervals directly assigned to the EIs and leaf node QCFs, i.e., the parameters based on the empirical input and assigned before the non-leaf node QCFs may be inferred) are expressed in terms of intervals within which the correct parameter values should lie. The width of the interval is proportional to the uncertainty of the domain experts or deduced from the standard deviation of the measurement-based input represented with probabilistic notions. In the latter case, the standard deviation indicates the accuracy of the measurements associated with each initially estimated parameter. Thus, the interval width may vary between the individual parameters. The representation of the estimates and their uncertainty is exemplified through an excerpt of a DV (with fictitious values) shown in Figure 7.

In addition to the quantifiable uncertainty associated with each initially estimated parameter, there may exist sources of uncertainty which are general for the context or the system itself, but to a lesser degree expressive or measurable. Examples include the presence of the aleatory uncertainty, the competence of the domain experts, data quality, statistical significance, etc. Such factors contribute to the overall uncertainty, but are (due to their weak expressiveness) not explicitly taken into account within the initially estimated EIs and the leaf node QCFs. Another reason for not accounting them within the intervals is because they are unavailable or may be biased at the individual parameter level. The domain experts may for example be subjective with respect to the above exemplified factors, or the tools for data acquisition may be incapable of providing the values regarding data quality, statistical significance, etc. Therefore, the context related uncertainty should, from an unpartial perspective (e.g., by a monitoring system or a panel, and based on a pre-defined rating), be expressed generally for all prior estimates.

Hence, we introduce the "confidence level" as a measure of the expected probability that the correct value lies within the interval assigned to a prior estimate. The confidence level is consistent and expresses the overall, uniform, context or system relevant certainty, in terms of a percentage. The confidence level regards the prior estimates only. The confidence level dictates the width of the intervals of the prior estimates, i.e., the certainty with which the exact value is within the interval assigned to a prior estimate. For example, a confidence level of 100% guarantees that the exact values lie within the intervals assigned to the prior estimates. Obviously, a requirement for increased confidence level will result in wider intervals of the prior estimates. In the case of Figure 7 the prior estimates are assigned with a confidence level of 90%. Let QCFs and EIs be represented by intervals of type x:

$$x = [\underline{x}; \overline{x}] = \{X \in [0; 1] : \underline{x} \leq X \leq \overline{x}\} \quad (2)$$

where $\underline{x}$ is the minimum estimated parameter value above which the exact value should (the term "should" is intentionally used in order to account for the confidence level of the prior estimates which is below 100%) lie, while $\overline{x}$ is the maximum parameter value below which the exact value should lie. Both $\underline{x}$ and $\overline{x}$ are represented by real numbers. The interval x of a prior estimate is assigned with the confidence level specified. Due to model completeness, EIs on the arcs pointing to the nodes with a common parent must satisfy:

$$\left(\Sigma_{i=1}^{I} \underline{x}_i\right) \leq 1 \wedge \left(\Sigma_{i=1}^{I} \overline{x}_i\right) \geq 1 \quad (3)$$

where i denotes index of an arc, I denotes the total number of the arcs with outspring from a common parent, and x_i denotes the interval estimate for the EI on arc i. That is, there must exist at least one subset of scalars from within each one of the intervals (representing EIs on the arcs to nodes with a common parent), whose sum is equal to 1.

B. Uncertainty propagation

The initial estimates are provided in the form of intervals with respect to a confidence level, as specified above. The propagation of the initially estimated intervals on the non-leaf node QCFs is given by the existing DV propagation algorithm (exemplified by Eq. 1 in Section II), the interval arithmetics [11], [12], and the algorithms for non-linear optimization [13], [14]. The result of the propagation is in the form of intervals of QCF values on the non-leaf nodes.

The confidence level itself is not propagated but only used in the context of the assignment of the initial estimates. Therefore, the confidence level is only associated with the initial estimates and not the inferred ones (non-leaf node QCFs). The confidence level does however affect the width of the inferred parameters through the width of the initial estimates. That is, since a requirement for a higher confidence level implies wider intervals of the initial estimates, the propagation will, as specified below, result in wider intervals on the non-leaf node parameters.

The only two interval arithmetic operations needed for propagation in a DV are addition and multiplication. In case of two intervals denoted by x and y (of the form given by Eq. 2), addition and multiplication are defined as:

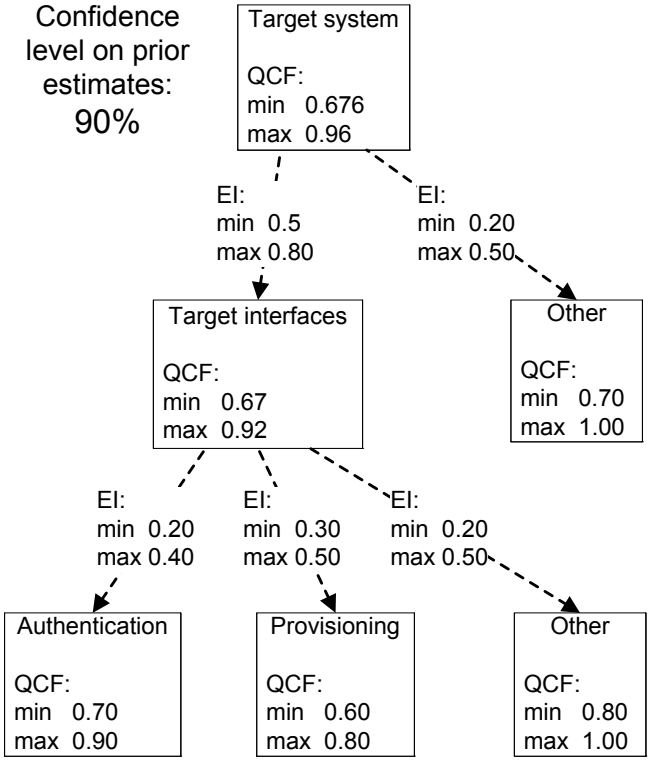

Figure 7. Excerpt of a DV with intervals and confidence level

$$x \circ y = [\underline{x} \circ \underline{y}; \overline{x} \circ \overline{y}] \quad (4)$$

Where $\circ$ denotes the operation symbol.

The optimization is necessary for obtaining the extreme values (the maximum and the minimum) of the interval of a parent node in the cases when several combinations (within the propagated intervals) give a sum of the EIs (on the arcs pointing to the immediate children) equal to 1. The scalar points (from within the intervals involved), which provide the extreme values, are identified by the non-linear optimization algorithms and then inferred to the parent node QCF in the form of an interval, according to the general DV propagation algorithm.

For a set of EI intervals whose total sum of the upper interval values is more than 1, there may be infinitely many combinations (the number of the combinations depends on the number of decimal digits, which the scalars from the intervals are represented with) of scalar points from within all the intervals, which together sum up to 1. Regardless of how many EIs (or nodes) there are, finding the min and the max values of the interval resulting from the propagation (sum of products of QCF and EI values associated with respectively the immediate children nodes and the arcs pointing to them) is a feasible optimization problem [14], [11]. Since the number of unknowns is equal to the number of equations involved, the only condition for the feasibility of the algorithm is the one expressed by Eq. 3.

Let $\underline{qcf}, \overline{qcf} \in [0;1]$ denote the interval limits of the QCFs on the immediate children and let $\underline{ei}, \overline{ei} \in [0;1]$ denote the EIs on their respective interconnecting arcs. We propose the utility functions for the inferred min and max for the intervals of the parent node QCFs, which are given by respectively:

$$\underline{QCF} \stackrel{def}{=} min\left\{\Sigma_{i=1}^{I} \underline{qcf}_i \cdot ei_i | \forall i \in I : \underline{ei}_i \leq ei_i \leq \overline{ei}_i \wedge \Sigma_{i=1}^{I} ei_i = 1\right\} \quad (5)$$

$$\overline{QCF} \stackrel{def}{=} max\left\{\Sigma_{i=1}^{I} \overline{qcf}_i \cdot ei_i | \forall i \in I : \underline{ei}_i \leq ei_i \leq \overline{ei}_i \wedge \Sigma_{i=1}^{I} ei_i = 1\right\} \quad (6)$$

I and i denote the same notions as in Eq. 3. The inference starts from the lowest internal nodes, and proceeds recursively upwards the tree.

The sensitivity of the inferred interval width of a dependent node, on the interval width of a dependee (node or arc), can be deduced by:

1) estimating the initial parameters and propagating them
2) obtaining the inferred interval width W of the selected dependent node
3) removing (or partially reducing) the interval width of the selected dependee D
4) obtaining the new inferred interval width W' of the dependent node
5) calculating the sensitivity S between the dependent node W and the dependee parameter D, with respect to uncertainty.

We define the sensitivity measure $S_{W,D}$ as:

$$S_{W,D} = (1 - \frac{W'}{W}) \quad (7)$$

In the context of predicting the quality characteristic, the natural choice of the dependent node will be the root node, which represents the quality characteristic that the DV is dedicated to, while the dependee will be a leaf node QCF or an EI. The QCF value on the root node will then represent the value of the quality characteristic of the system. The dependee is subject to the initial estimation. Therefore, the uncertainty of a dependee may be directly adjustable (for example, by reducing interval width due to added input). The sensitivity value can be obtained prior to selecting the candidate parameters for uncertainty reduction through added input. The obtained value of sensitivity (defined by Eq. 7) can in such a case be considered in relation to the effort needed for acquisition of the additional input. That is, higher sensitivity justifies putting more effort in acquiring additional input in order to decrease uncertainty of the dependee (and thus dependent) node.

C. The uncertainty propagation in practice

Currently, we run the optimization in *Matlab*, where the utility function is, based on the DV propagation model exemplified by Eq. 1, defined as the sum of products of the QCF and EI intervals related to the immediate children nodes. The constraints of the utility function are:

- all QCF intervals involved,
- all EI intervals involved, and

- $\Sigma_{i=1}^{I} ei_i = 1$ (where i denotes an arc, I is the total number of the arcs pointing to the nodes with the common parent under consideration, and ei_i is a variable representing the EI value on the arc i). This constraint ensures the model completeness.

The minimum of the inferred interval is obtained from the utility function, while the maximum of the inferred interval is obtained by inverting the sign on the left hand side of the utility function and re-running the non-linear optimization algorithm. The *Target interfaces* and *Target system* nodes in Figure 7 are examples where such an algorithm had to be run in order to obtain the propagated intervals. In the case of *Target interfaces*, the utility function is specified in *Matlab* as:

```
function f = objfun(x,y)
f = x(1)*x(2)+x(3)*x(4)+x(5)*x(6);
```

Where x(1), x(3) and x(5) represent the EI values on the arcs pointing to the *Authentication*, *Provisioning* and *Other* nodes, respectively; while x(2), x(4) and x(6) represent the QCF values on the *Authentication*, *Provisioning* and *Other* nodes, respectively.

The related nonlinear inequality constraints representing the max and the min interval values of each respective variable specified above are defined in *Matlab* as:

```
c = [-x(1) + 0.2; x(1) - 0.4; -x(2) + 0.7; x(2) - 0.9;
     -x(3) + 0.3; x(3) - 0.5; -x(4) + 0.6; x(4) - 0.8;
    -x(5) + 0.2;  x(5) - 0.5; -x(6) + 0.8; x(6) - 1.0];
```

The nonlinear equality constraint specifying that the sum of the EIs has to equal to 1, is defined in *Matlab* as:

```
ceq = [x(1) + x(3) + x(5) - 1];
```

The optimization algorithm is run by the following command in *Matlab*:

```
x0 = [0,0,0,0,0,0]; % Make a starting guess at the solution
options = optimset('LargeScale','on');
[x, fval] = ...
fmincon(@objfun,x0,[],[],[],[],[],[],@confuneq,options)
```

Providing the following result, where the values in the vector x specify the scalar points within the intervals x(1)-x(6), which yield the min value 0.67 of the utility function:

```
x = 0.3000  0.7000  0.5000  0.6000  0.2000  0.8000
fval = 0.6700
```

The max of the inferred interval is specified in *Matlab* by changing the sign of the above shown utility function to:

```
f = -(x(1)*x(2)+x(3)*x(4)+x(5)*x(6));
```

and re-running the command from above. The output obtained is:

```
x = 0.2000  0.9000  0.3000  0.8000  0.5000  1.0000
fval = 0.9200
```

where the values in the vector x specify the scalar points within the intervals x(1)-x(6), which yield the max value of the utility function, namely 0.92.

The propagation results are displayed in Figure 7. We see that the scalar points of the optimization output are in accordance with the Eq. 5 and Eq. 6.

D. Uncertainty analysis

Statistical analysis of measurements performed prior to model fitting and sensitivity analysis performed in relation to the application of prediction models, require a toolset for analysis of the data sets represented by intervals. The analysis of the central tendency measures of the interval-based estimates relies on the existing fully defined interval arithmetics and interval statistics [15]. Both can, in their existing well-established form, be directly applied in our context.

Apart from the summation and the multiplication presented by Eq. 4, the elementary interval arithmetic functions addition and multiplication (given two intervals denoted by x and y, both of the form given by Eq. 2) include subtraction and division:

$$x - y = [\underline{x} - \underline{y}, \overline{x} - \overline{y}] \tag{8}$$

$$x \div y = [\underline{x}, \overline{x}] \cdot [1/\overline{y}, 1/\underline{y}], \text{ as long as } 0 \notin y. \tag{9}$$

Arithmetic mean is given by:

$$\left[\tfrac{1}{I}\Sigma_{i=1}^{I}\underline{x}_i, \tfrac{1}{I}\Sigma_{i=1}^{I}\overline{x}_i\right]. \tag{10}$$

For geometric mean, harmonic mean, weighted mean, and median, see [15]. Since no two data values are likely to be the same at infinite precision, mode does not generalize to a useful summary for data sets containing interval values. Instead, [15] proposes a substitute statistic, which identifies the places where most values in the data set overlap.

For problems with large sample sizes, computing variance of the interval data is an NP-hard problem. The algorithms for calculating variance presented in [15] solve the issue of infeasibility and make practical calculations of the needed interval statistics.

The standard deviation σ of an interval can be computed immediately from the variance *var* by taking its square root:

$$\sigma = [\underline{\sigma}, \overline{\sigma}] = \sqrt{var} = \left[\sqrt{\underline{var}}, \sqrt{\overline{var}}\right]. \tag{11}$$

Interval statistics for interquartile range, skewness, confidence intervals, regression fitting, maximum likelihood methods, as well as inferential interval statistics are thoroughly presented in [15]. In addition, [15] provides guidance regarding identification of outliers, trade-off between sample size and precision, handling of measurement uncertainty, handling of dependencies among the sources of uncertainty (correlation and covariance) and accounting for incertitude.

IV. WHY OUR SOLUTION IS A GOOD ONE

This section argues that the approach presented above fulfills the success criteria defined in Section II. Each one of the six criteria is considered in a dedicated subsection.

A. Criterion 1

The interval-based approach extends the DV parameters with the notions of interval widths and confidence level. Both interval width and confidence level are based on

fairly intuitive and simple definitions. Hence, the approach should be relatively easy for the domain experts to use and understand, regardless of the degree of their formal background. The simplicity also makes it less prone to unstable over-fitting, as well as bias or inaccuracy of the estimations.

B. Criterion 2

The interval width can be selected at the individual prior estimate level, thus allowing adjustment of granularity of the uncertainty representation. The number of the decimal digits used in estimation and propagation is unlimited.

C. Criterion 3

The domain expert judgements are provided directly in terms of intervals with a confidence level. However the measurement-based input may come in terms of statistical notions.

Given that the measurement-based input is normally distributed, the interval end points can be calculated as [16]:

$$\mu \pm t(1-conf, n-1)\sigma\sqrt{\tfrac{1}{n}+1} \quad (12)$$

where $t(1-conf, n-1)$ is the two-tailed value of the Student's t-distribution for the confidence level $1-conf$ and $n-1$ degrees of freedom, $\mu \in [0;1]$ is the mean value, σ is the standard deviation of the measurements and n is the number of measurements. The "1" term inside the square root describes the spread of the measurement accuracy, while the "1/n" term describes the spread of the mean measurement accuracy. When n is high, there will be almost no uncertainty about the mean measurement accuracy, but the spread of the measurement accuracy may still be large. One can express both QCFs and EIs in this manner (for the relationship between the DV parameters and the measurements, see [2]), while requiring that Eq. 2 and Eq. 3 are satisfied. Alternatively, one can represent the QCF values in this manner, and the EI value of each related arc as a probability $p \in [0;1]$, while enforcing $\sum p = 1$ for all nodes having a common parent. Thus, both kinds of input are transformable to intervals, which then can be propagated as defined in Section III and exemplified below.

D. Criterion 4

A consequence of the inequality and equality constraints is that all the inferred values will lie within the interval [0;1]. In addition, the normalized quality characteristic metric is defined so that all possible values always must lie within this interval. Moreover, the propagation algorithm calculates both the upper and the lower extreme values. As a result, the inferred prediction is an interval within which the exact (factual) value should lie. Two aspects are hindering from guaranteeing that the factual value lies within the inferred interval:

1) the confidence level with which the prior estimates are provided, and
2) the aleatory uncertainty, which unless accounted for in the confidence level, is not quantified within the intervals.

	Prior estimates 90% conf. level			Propagated
	QCFs	EIs	QCFs and EIs	QCFs
Count	38	47	85	10
Max	0.05	0.15	0.15	0.0645
Min	0.00	0.00	0.00	0.0141
Avg	0.027	0.02	0.025	0.0366
StDev	0.0199	0.023	0.022	0.014

Table I
SUMMARY OF THE INTERVALS APPLIED ON A REAL DV STRUCTURE

E. Criterion 5

The interval-based approach has also been tested by providing example values of estimates and their uncertainty on a real DV structure. The DV structure was originally used in a feasibility study of the PREDIQT method [2], performed on an extensive, real system. The uncertainty estimates were straight-forward to provide by referring to the definition of the rating of the quality characteristic and expressing the estimates in terms of intervals. The interval width was mostly subject to observability of the parameter and existence of relevant historical input. The DV consisted of 38 leaf nodes, 9 internal nodes and 1 root node. The number of EIs on the arcs was 47. Thus, the number of initial (empirical input-based) estimates was 85, in this case. All initial estimates were expressed with intervals of reasonable and varying widths, within 90% confidence level. Once the initial estimates were in place, the propagation was quick and straightforward.

Table I summarizes the intervals applied. Each column lists the number of elements, the maximum interval width, the minimum interval width, the average interval width and the standard deviation of the interval width. The first two columns present the values for the initial estimates of the leaf node QCFs and all the EIs, respectively. The third column presents the values for the initial estimates of both the leaf node QCFs and all the EIs. The last column presents the results for the propagated QCFs (on the internal nodes and the root node). The resulting interval width of the root node QCF was 0.032. Given the attempt to provide as realistic and as variable interval widths of the initial estimates as possible, the example should be an indication of the expected findings in similar settings. Note that, while the interval widths reflect the expected uncertainty, all values assigned to parameter estimates are fictitious, due to their confidentiality. The obtained root node interval width can be considered as a promising result, since the predictions are still likely to be associated with limited and acceptable uncertainty.

To test impact of uncertainty elimination on one leaf node (a child node of the root node) on the above presented DV, its QCF was changed from [0.90;0.95] to [0.925;0.925]. The

resulting interval width of the root node QCF became 0.0295 and the value of Eq. 7 became 0.081. Note that these values, too, are based on fictitious input, due to confidentiality of the actual initial estimates.

In a real-life setting, not all the estimates will be expressed with uncertainty, since some of the nodes have no impact or no uncertainty. The evaluation of the above mentioned feasibility study showed that the uncertainty of the input and the deviations between the PREDIQT-based and the empirical predictions are relatively low. The experience from the feasibility study is that the interval widths would be quite small. Most of the nodes of the DVs were placed on the second or the third level, which considerably limits the vertical propagation of uncertainties.

Reducing the confidence level and conducting further model fitting (through additional input) are the obvious counter-measures when the inferred values are too uncertain. The candidate parameters for reduction of uncertainty can be identified by using the sensitivity measure proposed in Section III in relation to the effort needed for the uncertainty reduction in question. Alternatively, a sensitivity analysis supported by charts and central tendency measures can be pursued in order to observe the impact that a reduction of uncertainty of the individual estimates would have on (the root node of) the DV.

F. Criterion 6

The analysis of the central tendency measures of the interval-based estimates relies on the existing fully defined interval arithmetics and interval statistics [15]. Both can, in their existing well-established form, be directly applied in our context. For arithmetic mean, geometric mean, harmonic mean, weighted mean, median, standard deviation and variance, see [15]. In addition, [15] provides guidance regarding identification of outliers, trade-off between sample size and precision, handling of measurement uncertainty, handling of dependencies among the sources of uncertainty (correlation and covariance) and accounting for incertitude.

V. WHY OTHER APPROACHES ARE NOT BETTER IN THIS CONTEXT

A ratio scale is a measurement scale in which a certain distance along the scale means the same thing no matter where on the scale we are, and where "0" on the scale represents the absence of the thing being measured. Statistical analysis and arithmetics are supported for the ratio scale. The ratio scale is in fact used in Section II. We may for example introduce uncertainty representation by defining fixed increments on the scale from 0 to 1, and relating their meaning to the quality characteristic rating. The input would have to be expressed in the form of the increments defined, and the uncertainty would per definition range half the way to the neighboring increments. Obviously, this is a special case of the interval approach where the increments and their granularity are frozen at the model (and not parameter) level. By using a ratio scale in the PREDIQT context, the schema of the increments would have to apply for the entire model (in order for the uncertainty propagation to be meaningful) rather than being adjustable at the parameter level. As a result, the schema of the increments may be either too coarse grained or too fine grained in the context of certain parameters. The variation of uncertainty between parameters would not be supported, thus violating criterion 2 from Section II.

The Dempster-Shafer structures [15] offer a way of representing uncertainty quantified by mass distribution functions. A mechanism for aggregation of such representation stored in distributed relational databases, is proposed by [17]. The Dempster-Shafer approach characterizes uncertainties as intervals with degrees of certainty (that is, sets of values with weights which add up to 1). It can be seen as a generalization of both interval analysis and probability theory. Weights of evidence are put on a collection of intervals and the structures may overlap. Implementing the Dempster-Shafer theory in our context would involve solving two issues: 1) sorting the uncertainties in the empirical input into a priori independent items of evidence, and 2) carrying out Dempster's rule computationally. The former one leads to a structure involving input elements that bear on different but related concerns. This structure can be used to make computations based on Dempster's rule feasible. Our solution is a special case of the Dempster-Shafer approach, where the intervals of the prior estimates have a general confidence level, and the structure of the DV allows for a linear propagation. The additional expressiveness that the Dempster-Shafer structures offer is not needed in our context, since the certainty is highly unlikely to vary across the fractions of the intervals. In fact, such a mechanism will, due to its complex representation on subsets of the state space, in the PREDIQT context only compromise the comprehensibility of the uncertainty representation and therefore the correctness of the input.

Bayesian networks (BNs) [18], [19] may represent both model uncertainty and parameter uncertainty. A BN is a directed acyclic graph in which each node has an associated probability distribution. Observation of known variables (nodes) allows inferring the probability of others, using probability calculus and Bayes theorem throughout the model (propagation). BNs can represent and propagate both continuous and discrete uncertainty distributions. BNs in their general form are however demanding to parameterize and interpret the parameters of, which violates our first criterion. This issue has been addressed by [20] where an analytical method for transforming the DVs to Bayesian networks is presented. It also shows that DVs, although easier to relate to in practice, are compatible with BNs. It is possible to generalize this transformation so that our interval-based approach is transformed to a BN before

a further BN-based analysis may be conducted. Such an extension would introduce several states on the BN nodes, and assign probabilities to each of them. In that manner, the extension would resemble the Dempster-Shafer structures. BNs in their general form do not score sufficiently on our criteria 1 and 5.

Fuzzy logic provides a simple way to draw definite conclusions from vague, ambiguous or imprecise information, and allows for partial membership in a set. It allows modeling complex systems using higher levels of abstraction originating from the analyst's knowledge and experience [21]. A fuzzy set is a class of objects with a continuum of grades of membership. Such a set is characterized by a membership function, which assigns to each object a grade of membership ranging between zero and one [22]. Using the fuzzy membership functions, a parameter in a model can be represented as a crisp number, a crisp interval, a fuzzy number or a fuzzy interval. In the fuzzy logic approach the algebraic operations are easy and straightforward, as argued and elaborated by [23]. The interval-based approach is a special case of the fuzzy approach, where only the crisp intervals are used as membership functions. The additional expressiveness that the overall types of the membership functions offer is in fact not needed in the PREDIQT context, since the increased complexity of the estimate representation would not contribute to the accuracy of the parameter values, but rather introduce misinterpretations and incorrectnesses in the input provision. The interpretation of the membership distributions and their correspondence to the practical settings in the PREDIQT context would be demanding.

Subjective logic [24] is a framework for reasoning, which consists of a belief model called *opinion* and set of operations for combining opinions. A single opinion π is uniquely described as a point $\{b, d, i\}$ in an "Opinion Triangle", where b, d and i designate belief, disbelief and ignorance, respectively. For each opinion, the three notions sum up to unity. The operations formally defined include: conjunction, disjunction, negation, consensus, recommendation and ordering. The subjective logic is suited for the domain expert judgements, but how the measurement-based input can be related to the concepts of the subjective logic, needs to be defined. Thus, applying the subjective logic in the PREDIQT context would increase the fulfillment of our second criterion beyond the needs, while degrading fulfillment of the third criterion.

Uncertainty representation in software development effort-estimation [25], [26] is most comparable to ours. However, they do not have as a strict criterion of propagation, and can therefore introduce different notions to the uncertainty representation.

It should be pointed out that the interval propagation based on the extreme values suffers from the so-called overestimation effect, also known as the dependency problem. The dependency problem is due to the memoryless nature of interval arithmetic in cases when a parameter occurs multiple times in an arithmetic expression, since each occurrence of an interval variable in an expression is treated independently. Since multiple occurrence of interval parameters cannot always be avoided, the dependency problem may cause crucial overestimation of the actual range of an evaluated function. A way to approach this issue is to use interval splitting [27], where the input parameter intervals are subdivided and the arithmetics are preformed on the subintervals. The final results are then obtained by computing the minimum of all lower bounds and the maximum of all upper bounds of the intermediate results. Skelboe [28] has shown that the results obtained from the interval splitting converge to the actual range if the width of the subintervals approaches zero. Our solution does not use interval splitting, as it would significantly increase complexity of the entire approach, thus compromising our first criterion.

The epistemic uncertainty is the crucial one in the context of PREDIQT and therefore given the main attention in our context. Being of a discrete nature, the epistemic uncertainty should, as argued in Section II, be handled by a purely possibilistic approach. The approaches mentioned in the remainder of this section focus to a high degree on the stochastic uncertainties, which makes them less suited in the PREDIQT context.

The ISO approach to handling measurement uncertainty [29] uses a probabilistic representation with normal distribution, and treats both aleatory and epistemic uncertainty equally. Such an approach however does not explicitly account for the notion of ignorance about the estimates, thus failing to intuitively express it.

A simulation mechanism, which takes into account both aleatory and epistemic uncertainty in an interval-based approach, is proposed by [30]. It concentrates on stochastic simulations as input for the interval estimates when significant uncertainties exist. Moreover, [15] proposes considering a hybrid approach comprising both probabilistic and interval representation, in order to account for both aleatory and epistemic uncertainty. Neither of these two approaches would in the the context of PREDIQT increase fulfillment of our success criteria. In fact, the systematic sources of uncertainty would not be represented more accurately, while comprehensibility would degrade.

A hybrid Monte Carlo and possibilistic method for representation and propagation of aleatory and epistemic uncertainty is presented by [31]. The method is applied for predicting the time to failure of a randomly degrading component, and illustrated by a case study. The hybrid representation captures the aleatory variability and epistemic imprecision of a random fuzzy interval in a parameterized way through α-cuts and displays extreme pairs of the upper and lower cumulative distributions. The Monte Carlo and the possibilistic representations are jointly propagated. The gap between the upper and the lower cumulative distributions

represents the imprecision due to epistemic variables. The possibility distributions are aggregated according to the so called Ferson method. The interpretation of the results in the form of limiting cumulative distributions requires the introduction of a degree of confidence directly connected with the confidence on the value of epistemic parameters. Compared to this approach, our solution is more comprehensible but less suited for handling the aleatory uncertainty. However, given our criteria, the former aspect outranges the latter one.

The approaches to uncertainty handling in other domains, such as weather forecasting [32], electricity demand forecasting [33], correlations between wind power and meteorological conditions [34], power system planning [35] and supply industry [36] are mainly based on probabilistic representations and stochastic simulations. They focus mainly on the aleatory uncertainty, which in the PREDIQT context is of secondary relevance.

Hence, given the criteria presented in Section II, the interval-based approach prevails as the most appropriate one in the PREDIQT context.

VI. Lessons learned

This section provides practical guidelines for obtaining the empirical input and reducing the uncertainty of estimates. Firstly, we elaborate on how the maximum acceptable uncertainty objective, that is, an acceptable threshold for uncertainty, can be characterized. Secondly, guidelines for obtaining the prior estimates are summarized. Lastly, means and measures for reducing uncertainty are outlined. The guidelines are based on the authors' experiences from industrial trials of PREDIQT on real-life systems [2], [3]. As such, the guidelines are not exhaustive but may serve as an aid towards a more structured process for uncertainty handling.

A. Characterizing the maximum acceptable uncertainty objective

The maximum acceptable uncertainty objective can to a certain degree be expressed through the confidence level, which is a measure of the expected probability that the correct value lies within the interval assigned to a prior estimate. However, the confidence level is merely concerned with the prior estimates although it indirectly influences the inferred DV estimates. Therefore, if the interval width of a specific non-leaf node is of major concern, it has to be specified directly as a part of the maximum acceptable uncertainty objective, by the stakeholders. Note however that there is still a correlation between the confidence level of the prior estimates and the inferred QCFs, that is, the uncertainty of an inferred QCF is expressed through both width of its interval, as well as the confidence level of the prior estimates which influence the QCF value of the non-leaf node in question.

Consequently, in the case of the prior estimates, the maximum acceptable uncertainty objective can be expressed through the confidence level, and will in that case give interval widths depending on the quality of the empirical input. In the case of the non-leaf node QCF values, the maximum acceptable uncertainty objective should be expressed in terms of both the confidence level of the prior estimates and the interval width of the parameters in question.

B. Obtaining the prior estimates

We recommend obtaining the leaf node QCFs of a subtree prior to obtaining the related EIs. The rationale for this is to fully understand the semantics of the nodes, through reasoning about their QCFs first. In estimating a QCF, two steps have to be undergone:

1) interpretation of the node in question – its contents, scope, rationale and relationship with the Design Models, and
2) identification of the relevant metrics from the Quality Model of the quality characteristic that the DV is addressing, as well as evaluation of the metrics identified.

QCF is the degree of fulfillment of a quality characteristic, with respect to the node in question. Normalization of the values of the above mentioned metrics and their degree of influence, results in a QCF value with an uncertainty interval assigned with respect to the pre-defined confidence level. Alternatively, rating of the characteristic (as formally defined by its Quality Model at the root node level) can be estimated directly with respect to the node under consideration, in order to provide its QCF value.

In estimating an EI, two steps have to be undergone:

1) interpretation of the two nodes in question, and
2) determination of the degree of impact of the child node on the parent node, with respect to the quality characteristic (defined by the Quality Model) that the DV is addressing. The value is assigned relative to the overall EIs related to the same parent node, and with a consistent unit of measure, prior to being normalized (in order to fulfill Eq. 2). The normalized EIs on the arcs from the same parent node have to fulfill Eq. 3, due to the requirement of model completeness.

Hence, EI is the dependency of the parent node on the child node. Estimation of the EI values between a parent node and its immediate children, results in intervals with respect to the pre-defined confidence level.

1) Questions to ask domain experts: The first step in the interaction between the analyst and the domain experts is to clarify the meaning of the node(s) under consideration, their respective rationales and the possible traces to the Design Models. Secondly, the analyst has to facilitate the estimation by reminding the domain experts of the quality characteristic definition – both the qualitative and the formal part of it.

When estimating a QCF the following question is posed: *"To what degree is the quality characteristic fulfilled, given*

the contents and the scope of the node?" The definition of the quality characteristic (interpretation and the metric) should be recalled.

When estimating an EI the following question is posed: *"To what degree does the child node impact the parent node, or how dependent is the parent node on child node, with respect to the quality characteristic that the DV is dedicated to?"* The definition of the quality characteristic provided by its Quality Model, should be recalled and the estimate is provided relative to the impact of the overall children nodes of this parent. Alternatively, an impact value is assigned using the same unit of measure on all arcs of the sub-tree, and normalized thereafter.

Once one of the above specified questions is posed, depending on the kind of the DV parameter, the domain expert panel is asked to provide the estimate with an interval so that the correct value is within the interval with a probability given by the confidence level. For EIs on the nodes having a common parent, it has to be validated that Eq. 3 is fulfilled.

Furthermore, discussions among the domain experts should be encouraged and all the estimates should be requested during a limited period of time (in the form of tightly scheduled meetings), in order to ensure relative consistency of the estimates. Additionally, for the purpose of the relative consistency of the estimates, the domain expert group should be diverse and representative. There should be continuity in a fraction of the group, and limited turnover between the different meetings. The turnover may however be advantageous for the purpose of the expertise at the different stages of the process of PREDIQT.

Apart from the domain expert judgements, the estimates are also based on measurements. When obtaining measurement-based input, we rely on a measurement plan which relates the practical measurements to the DV parameters and the quality notions. The Goal/Question/Metric [37], [38], [39] approach and the ISO 9126 product quality standard [5] are particularly useful for deducing such relationships. The overall literature on software measurement is extensive [40], [41], [42], [43] and provides useful guidelines for obtaining the measurement-based input.

2) Use of Quality Models: Quality Models are used as a reference in estimation of each prior estimate. The Quality Models assist the domain experts in selecting and evaluating the relevant metrics. The metrics also provide a basis for defining the measurements. The decomposition of the Quality Models is however only based on indicators whose overlaps and degrees of impact on the characteristic may vary. The composition of the degrees of relevance of the various indicators is therefore left to the analyst or the domain experts to determine in the case of each estimate.

3) Use of Design Models: Design Models specify the target of the analysis in terms of scope and the contents. The Design Models serve as a reference for common understanding of the system, prior to and during the estimation. In addition, the appropriate parts of the DVs are traced to the elements of the Design Models, making the contents and the scope of the DV elements more explicit.

4) Determining the uncertainty value: The uncertainty value of a prior estimate is determined through the interval width based on the pre-defined confidence level. In the case of the measurement-based input, the transformation to an interval is presented in Section IV. In that context, confidence level will reflect the data quality (that is, the validity of the measurements).

In the case of the domain expert judgements, however, the interval width is agreed upon by the domain expert panel, while the validity of the panel (that is, mainly representativeness and statistical significance of its composition) is reflected through the confidence level. This ensures consistency of the confidence level in the case of the expert judgements.

In order to also ensure a consistent confidence level in the case of the measurements (where data quality may vary among the measurements related to the different DV estimates), the confidence level can be kept consistent by compensating for the possible variations through the interval width. The relationship between the confidence level and the interval width is however not formalized beyond the fact that the confidence level denotes the probability of the correct value of the estimate lying within the interval.

C. Reducing uncertainty

Since we only consider the epistemic uncertainty, there exist means and measures that can be used to reduce it. The difficulty of reducing uncertainty lies in addressing the unknown sources of uncertainty, which are not explicitly expressed in the estimates. This is however not a major issue in the case of the epistemic uncertainty.

The rest of this section provides guidelines for uncertainty reduction from the different perspectives: process, model granularity, measurement-based input and expert judgements.

1) Process related measures: Among the process related measures are:

- access to the necessary documentation
- access to measurement facilities
- involvement and composition of the domain expert panel in all phases of the process
- common understanding of the modeling language and the terminology
- sufficient understanding of the PREDIQT method, particularly the models (by the domain experts and the analyst)
- use of known notations and modeling frameworks
- use of standards where appropriate
- user-friendly tool support with structured process guidance
- reuse of the existing models where appropriate.

The rationale for these measures is a more structured process which provides the sufficient input and leads towards a harmonized understanding of the models. For a more detailed elaboration of the process related measures, see [3].

2) Granularity of the models: Quality of a model-based prediction is, once the prediction models are developed, subject to the granularity of the models. Increased granularity of all prediction models will potentially decrease uncertainty.

In case of Quality Models, finer granularity can be achieved by further formalization and decomposition of the quality characteristics. In case of Design Models, the more detailed diagrams and traces among them are a means of addressing granularity.

In the case of the DVs, additional traceability of the actions and rationale, as well as increased traceability between DV model elements and the Design Models, will increase the precision and reduce uncertainty. Particularly, the following should be documented during the DV development:

- assumptions
- rationale
- relationships or traces to the Design Models
- traces to the relevant quality indicators and contributions of the relevant quality indicators
- interpretations of the prior estimates
- the supporting information sources (documents, measurement, domain experts) used during the development of DV structure and estimation of the parameters.

3) Quality of measurement data: Increase of validity of the measurement data will directly increase the confidence level. This may be achieved by increasing the statistical significance of the measurements in terms of relevance and amount of the measurement-based input.

4) Quality of expert judgements: The expert judgements are subject to understandability and granularity of the prediction models, composition of the expert panel (representativeness, number of participants, their background and interests), setup and approach to the estimate acquisition. Discussion should be facilitated and possible interest conflicts should be addressed.

VII. Conclusion and Future Work

Our earlier research indicates the feasibility of the PREDIQT method for model-based prediction of impacts of architectural design changes on system quality. The PREDIQT method produces and applies a multi-layer model structure, called prediction models, which represent system design, system quality and the interrelationship between the two. A central part of the prediction models are the DVs, which are parameterized in terms of fulfillment of quality characteristics and impacts among the elements, with respect to the quality characteristics. The DV elements are representations of architectural design or quality, which are partially traceable to the underlying Design Models and Quality Models. Due to its empirical nature, input into the DVs is associated with uncertainty. By handling the uncertainty in the DVs, quality of the prediction models and accuracy of the predictions are made explicit, thus indicating which changes are predictable and whether further model fitting is needed.

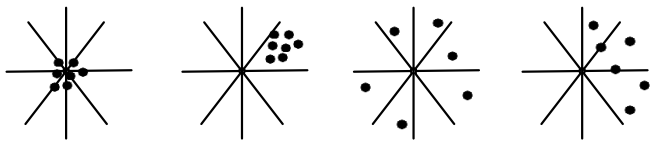

Figure 8. 1: Accuracy; 2: Bias; 3: Precision

Based on a set of criteria identified with respect to the PREDIQT method, we have proposed and evaluated an approach to uncertainty handling in the DVs. The approach relies on intervals with a confidence level, and covers representation, propagation and analysis of the DV parameters and their respective uncertainty estimates. The interval-based approach allows comprehensible representation of uncertainty on all kinds of parameters, with the needed accuracy. Estimation, propagation and analysis in the interval-based approach are scalable and efficient. The interval arithmetics, the algorithms for non-linear optimization, and the statistical analysis of intervals are already fully established and can be applied in the PREDIQT context in their existing forms. The evaluation argues for the correctness and practical usefulness of our approach, as well as its outranging appropriateness relative to the alternative uncertainty handling approaches.

The approach is entirely compliant with the existing version of the PREDIQT method. Based on empirical trials of PREDIQT, we have provided guidelines for use of the uncertainty handling approach in practice. The guidelines address the ways of obtaining the empirical estimates as well as the means and measures for reducing uncertainty of the estimates.

Further work will address analysis of the prediction accuracy, that is the deviation between the predicted and the actual quality characteristic values. The notions of magnitude of average deviation AD [2], balanced relative error BRE [44] and hit rate (i.e., the percentage of the correct values lying within the predicted intervals) can be used as measures of prediction accuracy. For an accurate prediction model, the hit rate should be consistent with the confidence level. The BRE allows analysis of bias and precision (see Figure 8) of the predictions. Thus, systematic and random variance of the prediction accuracy can be distinguished in a meta analysis of our uncertainty handling approach. The prospects of further work also include additional empirical evaluations of practical usefulness and accuracy of the approach. Moreover, identifying and categorizing the variables that impact the uncertainty of the estimates, is important for improving uncertainty management.

ACKNOWLEDGMENT

This work has been conducted as a part of the DIGIT (180052/S10) project funded by the Research Council of Norway, as well as a part of the NESSoS network of excellence funded by the European Commission within the 7th Framework Programme.

REFERENCES

[1] A. Omerovic and K. Stølen, "Interval-Based Uncertainty Handling in Model-Based Prediction of System Quality," in *Proceedings of Second International Conference on Advances in System Simulation, SIMUL 2010*, August 2010, pp. 99–108.

[2] A. Omerovic, A. Andresen, H. Grindheim, P. Myrseth, A. Refsdal, K. Stølen, and J. Ølnes, "A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality," SINTEF A13339, Tech. Rep., 2010.

[3] A. Omerovic, B. Solhaug, and K. Stølen, "Evaluation of Experiences from Applying the PREDIQT Method in an Industrial Case Study," SINTEF, Tech. Rep. A17562, 2011.

[4] "Excel Help and How-to," accessed: June 7, 2011. [Online]. Available: http://office.microsoft.com/en-us/excel-help

[5] International Organisation for Standardisation, "ISO/IEC 9126 - Software engineering – Product quality," 2004.

[6] J. Rumbaugh, I. Jacobson, and G. Booch, *Unified Modeling Language Reference Manual*. Pearson Higher Education, 2004.

[7] A. Omerovic, A. Andresen, H. Grindheim, P. Myrseth, A. Refsdal, K. Stølen, and J. Ølnes, "A Feasibility Study in Model Based Prediction of Impact of Changes on System Quality," in *Proceedings of International Symposium on Engineering Secure Software and Systems ESSOS10*, vol. LNCS 5965. Springer, 2010, pp. 231–240.

[8] A. D. Kiureghiana and O. Ditlevsenb, "Aleatory or epistemic? Does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[9] S. Parsons, "Current Approaches to Handling Imperfect Information in Data and Knowledge Bases," *IEEE Trans. on Knowl. and Data Eng.*, vol. 8, no. 3, pp. 353–372, 1996.

[10] A. Omerovic, A. Karahasanovic, and K. Stølen, "Uncertainty Handling in Weighted Dependency Trees – A Systematic Literature Review," in *Dependability and Computer Engineering: Concepts for Software-Intensive Systems*, L. Petre, K. Sere, and E. Troubitsyna, Eds. IGI, 2011, accepted to appear as a chapter in the book.

[11] R. B. Kearfott, "Interval Computations – Introduction, Uses, and Resources," *Euromath Bull*, vol. 2, pp. 95–112, 1996.

[12] V. Kreinovich, J. G. Hajagos, W. T. Tucker, L. R. Ginzburg, and S. Ferson, "Propagating Uncertainty through a Quadratic Response Surface Model," Sandia National Laboratories Report SAND2008-5983, Tech. Rep., 2008.

[13] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[14] A. R. Ravindran, *Operations Research and Management Science Handbook*. CRC Press, 2008.

[15] S. Ferson, V. Kreinovich, J. Hajagos, W. Oberkampf, and L. Ginzburg, "Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty," Sandia National Laboratories Report SAND2007-0939, Tech. Rep., 2007.

[16] T. Wonnacott and R. Wonnacott, *Introductory Statistics*. Wiley, 1990.

[17] B. Scotney and S. McClean, "Database Aggregation of Imprecise and Uncertain Evidence," *Inf. Sci. Inf. Comput. Sci.*, vol. 155, no. 3-4, pp. 245–263, 2003.

[18] M. Neil, N. Fenton, and L. Nielsen, "Building Large-Scale Bayesian Networks," *Knowledge Engineering Rev.*, vol. 15, no. 3, pp. 257–284, 2000.

[19] D. Heckerman, A. Mamdani, and W. M. P., "Real-World Applications of Bayesian Networks," *ACM Communications*, vol. 38, no. 3, pp. 24–26, 1995.

[20] A. Omerovic and K. Stølen, "Simplifying Parametrization of Bayesian Networks in Prediction of System Quality," in *Proceedings of Third IEEE International Conference on Secure Software Integration and Reliability Improvement*. IEEE, 2009, pp. 447–448.

[21] D. P. Weber, "Fuzzy Fault Tree Analysis," in *Proceedings of the 3rd IEEE Conference on EEE World Congress on Computational Intelligence*. IEEE, 1994, pp. 1899–1904.

[22] L. A. Zadeh, "Fuzzy Sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[23] P. V. Suresh, A. K. Babar, and V. Raj, "Uncertainty in Fault Tree Analysis: A Fuzzy Approach," *Fuzzy Sets Systems*, vol. 83, no. 2, pp. 135–141, 1996.

[24] A. Jøsang, "Artificial Reasoning with Subjective Logic," in *Proceedings of the 2nd Australian Workshop on Commonsense Reasoning*. Australian Computer Society, 1997.

[25] S. Grimstad and M. Jørgensen, "Inconsistency of Expert Judgment-Based Estimates of Software Development Effort," *Journal of Systems and Software*, vol. 80, no. 11, pp. 1770–1777, 2007.

[26] T. M. Gruschke and M. Jørgensen, "Assessing Uncertainty of Software Development Effort Estimates: Learning from Outcome Feedback," *ACM Transactions on Software Engineering and Methodology*, vol. 17, no. 4, pp. 20–35, 2008.

[27] S. Majumdar, L. Johannes, G. Haring, and R. Ramadoss, "Characterization and Analysis of Software and Computer Systems with Uncertainties and Variabilities," in *Proceedings of Performance Engineering, State of the Art and Current Trends*, vol. LNCS 2047. Springer, 2001, pp. 202–221.

[28] S. Skelboe, "Computation of Rational Interval Functions ," *BIT Numerical Mathematics*, vol. 14, no. 1, pp. 87–95, 1974.

[29] International Organisation for Standardisation, "Guide to the Expression of Uncertainty in Measurement," 1993.

[30] O. G. Batarseh and Y. Wang, "Reliable Simulation with Input Uncertainties using an Interval-Based Approach," in *Proceedings of the 40th Conference on Winter Simulation*, 2008, pp. 344–352.

[31] P. Baraldi, I. C. Popescu, and E. Zio, "Predicting the Time To Failure of a Randomly Degrading Component by a Hybrid Monte Carlo and Possibilistic Method," in *Proceedings of International Conference on Prognostics and Health Management 2008*. IEEE, 2008, pp. 1–8.

[32] T. N. Palmer, "Predicting Uncertainty in Forecasts of Weather and Climate," *Rep.Prog.Phys.*, vol. 63, pp. 71–116, 2000.

[33] J. W. Taylor and R. Buizza, "Using Weather Ensemble Predictions in Electricity Demand Forecasting," *International Journal of Forecasting*, vol. 19, no. 1, pp. 57–70, 2003.

[34] M. Lange and D. Heinemann, "Accuracy of Short Term Wind Power Predictions Depending on Meteorological Conditions," in *Proceedings of Global Windpower Conference*, 2002.

[35] A. P. Douglas, A. M. Breipohl, F. N. Lee, and R. Adapa, "Risk due to Load Forecast Uncertainty in Short Term Power System Planning," in *IEE Transactions on Power Systems*, vol. 13, no. 4. IEEE, 1998, pp. 1493–1499.

[36] K. L. Lo and Y. K. Wu, "Risk Assessment due to Local Demand Forecast Uncertainty in the Competitive Supply Industry," in *IEE Proceedings on Generation, Transmission and Distribution*, vol. 150, no. 5. IEEE, 2003, pp. 573–581.

[37] V. R. Basili, "Software Modeling and Measurement: the Goal/Question/Metric Paradigm," University of Maryland, Tech. Rep. TR-92-96, 1992.

[38] V. Basili, G. Caldiera, and H. Rombach, "The Goal Question Metric Approach," *Encyclopedia of Software Engineering*, 1994.

[39] N. E. Fenton and S. L. Pfleeger, *Software Metrics: A Rigorous and Practical Approach*. PWS Publishing Co., 1998.

[40] C. Ebert, R. Dumke, M. Bundschuh, A. Schmietendorf, and R. Dumke, *Best Practices in Software Measurement*. Springer Verlag, 2004.

[41] C. Ebert and R. Dumke, *Software Measurement*, 1 ed. Springer-Verlag Berlin Heidelberg, 2007.

[42] S. H. Kan, *Metrics and Models in Software Quality Engineering*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., 1994.

[43] J. C. McDavid and L. R. L. Hawthorn, *Program Evaluation and Performance Measurement : An Introduction to Practice*. Sage Publications, 2006.

[44] J. Armstrong, *Long-Range Forecasting*. Wiley, 1985.

Revisiting Urban Taxis: Optimal Real Time Management And Performance Appraisal By A Discrete Event Simulation Model

Eugénie Lioris, Guy Cohen, Arnaud de La Fortelle

***Abstract*—Cities are complex spatial structures presenting dense economical and cultural activities where an efficient transportation system is essential to meet all the challenges encountered when considering a competent industrialization, massive production, other activities and an improved quality of life. Based upon the fact that car travel is still the consumer's first choice, we are interested in the challenging idea of creating an intelligently administered new system called "Collective Taxis". The service quality provided will be comparable with that of conventional taxis (system operating with or without reservations, door-to-door services, well adapted itineraries following the current demand, controlling detours and waits, etc.), with fares set at rates affordable by almost everyone, simply by utilizing previously wasted vehicle capacity. With the aim of achieving optimal functioning of such an elegant but also complex structure, a made to measure discrete event computer simulator has been developed as a tool to evaluate and fine tune real time decision algorithms, optimize resources, study the influence of various factors pertaining to demand (level, geometry) upon performances and the threshold of profitability. The aim of this paper is to present such a methodology and illustrate it with results corresponding to fictitious but realistic data for a city like Paris.**

***Keywords*-Intelligent transportation system, discrete-event simulation, Monte Carlo simulation, Poisson processes, routing algorithms, Pareto optimality, performance evaluation, parameter optimization, queueing network model.**

I. Merits Of Collective Taxis

A preference for automobile use in urban areas is obviously related to a variety of advantages, regarding comfort, convenience and speed when conditions permit. Many attempts have been made to reduce individual automobile dependency with car sharing, car pooling (such structures often accompanied by poor spatial distribution, greater concentration on high-demand destinations and additional constraints requiring users to return vehicles to specific stations) and even schemes to combine the use of private and public transport, but despite all the encouragement given, they remain only partial solutions and cannot be considered to be realistic transport alternatives geared to a broad panel of customers.

Conventional taxis are potentially an alternative to private vehicles. However their high cost prohibits daily use for most commuters, and fuel shortages, parking restrictions, difficult road conditions and pollution issues all contribute to reducing the operational efficiency of the classical taxi system, which has become outmoded and in need of development. A collective taxi system intelligently associating more than one passenger with each vehicle and operating in urban areas to provide a comparable service quality to conventional taxis at an affordable price could be an interesting solution to the growing transportation problem. Indeed, it would offer a service quality equivalent to individual cars without the need to drive in a hazardous environment and the consternation of being unable to find available parking on arrival. Additional advantages in terms of cost, environmental impacts and traffic conditions can also be expected by simply raising the occupancy rate of vehicles in cities and dynamically optimizing the vehicle itineraries.

The idea of collecting several clients with the same vehicle has already a long history, especially under the terminology of "demand responsive transport" or "dial a ride", which mostly implies systems operating with preliminary reservations, but such systems are often geared to particular groups of customers (disabled persons for travels between home and health centers, passengers between hotels and airports, etc.), and also frequently confined to particular areas or itineraries. Our aim here is to study the most open and less constrained transportation system operating in a whole city, and possibly its suburbs, with or without preliminary reservations, and addressed to all categories of customers, from the usual driver of a private car, to the aged person who can no longer drive in crowded cities. But such an open and flexible system is very difficult to operate, and the need to first assess its performances and economic profitability, which depend on the care with which it is designed and operated, and also on the characteristics of the potential demand, should be obvious.

The main goal of this paper, a preliminary version of which was presented at a IARIA conference [1], is to propose a simulation tool with such an objective in mind, and to demonstrate how it can be used to answer various questions. At this stage of our study, no real life implementation has been achieved yet, and no real data have been collected. Numerical experiments have been conducted with fictitious but hopefully realistic data inspired by a city like Paris. Hence, the reader is not asked to adhere to any definite conclusion, but he is invited to see how the proposed tool can be used to shade some light on various issues raised by such a system (actually, at this stage, only the so-called decentralized mode of operation, that is, without preliminary reservations, have been studied).

E. Lioris is with Imara Team, INRIA, Le Chesnay, France and with CERMICS, École des Ponts-ParisTech, Marne-La-Vallée, France, jennie.lioris@cermics.enpc.fr

G. Cohen is with CERMICS, guy.cohen@mail.enpc.fr

A. de La Fortelle is with CAOR, École des Mines-ParisTech and with Imara Team-INRIA, arnaud.de_la_fortelle@mines-paristech.fr

The rest of this paper is comprised of six sections: Section II presents the three modes of operation we have in mind for the exploitation of collective taxis to provide a door-to-door service to clients wishing to travel with and/or without reservations. The simulation tool we have designed is ready for those three modes, but as already said, only the mode without reservations has been investigated in some detail yet. Section III justifies the need of simulations for learning the system behavior. Moreover it discusses the simulation technique employed, presenting the different entities composing the system and their relationship. Next, the architecture of the constructed simulator is discussed, its different parts being introduced as well as the necessary input and the results issued. The different possibilities of performing a simulation are established according to the needs of the study as well as the exploitation of the results. Section IV examines the optimization problems encountered in the real time management and in the decentralized management of each vehicle, and then it provides a brief description of the utilized algorithm to decide upon client acceptance.

In §V, after introducing the necessary input data, it is demonstrated how the simulation tool allows for a very sharp microscopic analysis and the possible detection of some abnormal functioning in some parts of the system, and how to remedy it. Section VI presents a more macroscopic (or statistical) analysis of the results and, based on it, a methodology for strategy selection and resource optimization. Finally, §VII concludes the study introduced in this paper and discusses the challenge of a future study on the centralized and mixed managements.

II. The System As We See It

The principle of associating more than one passenger with each taxi already exists in many countries but the system is not always optimized. In general with such systems of so-called *shared taxis*, it is mostly the driver and/or, less frequently, the existing passengers who decide if a new prospective client will be accepted, what the vehicle itinerary will become, and what the fare should be. This type of transport system mostly exists in developing countries [2], [3, pp. 472–474]. In general the taxi routes start and finish in central town locations such as taxi ranks, lorry parks, train or bus stations etc. [2], and invariably the principal routes served are subject to common delays.

However the system has started to be developed in the US (New York [4], San Francisco [5], Los Angeles [6], [7]), Europe: Netherlands [8], Brussels [9], UK [10], Germany [11], etc.

In the above structures, the proposed services do require advanced reservations; some suggest fixed or semi-fixed routes, whilst others offer door-to-door services. The majority covers a large but still limited area (the suburbs of the town are not always included, others cover specified avenues, etc.). Sometimes the service hours are restricted (late night journeys are often rare and sometimes impossible). Others combine public transport with a taxi service where, depending on passenger demand, the bus driver calls a company to reserve a vehicle and the pick up point will be at a bus stop. Others restrict their services to females, the elderly or people with restricted mobility. Often it is the driver and/or the clients who decide if they want to enter and what the cost should be.

Demand Responsive Transport (DRT) is getting increasingly popular and the need to provide an optimized service clearly stands.

Unlike the existing shared taxi systems we have previously referred to, for which studies and/or implementation are conducted principally for systems requiring prior reservations, and fixed or only slightly variable routes and combined taxi/bus services [12], [13], [14], [15], we are interested in the optimization of a more flexible and open system. The DRT structure envisaged should impose the minimum of constraints to both clients and drivers whilst offering a service quality comparable to conventional taxis in terms of detours, initial waiting times. More precisely we are aiming at a system providing a door-to-door service with minimal waiting times, independence and optimized itineraries (almost direct itineraries) at a low cost for both vehicles and clients.

Given the novelty of the system, it is hard to predict the proportion of potential customers it will attract in a given area, and this demand will anyway depend on the service quality offered and on the fares. Hence, our approach is to propose an optimized configuration for any demand level and geometry. In other words the outcome of this study is to provide a tool to construct the *offer curve* (in terms of performances and costs) as a function of demand. Specialists in the field of transportation should be able to return the *demand curve* by comparison with other transportation means in a given city: as always in Economy, the intersection of both curves will determine the part this system may take.

Three operating modes of the “Collective Taxis” are envisaged:

- The decentralized management where clients appear randomly in the network seeking a vehicle for an immediate departure. Such a structure requiring no prior bookings was initially studied by [16]; our study here is an advancement of that work to include matters of simulation techniques, results and conclusions about its relevance compared to the next modes of operation in any specific situation.
- The centralized management dealing with clients who must make an advanced reservation of a seat in a vehicle.
- The mixed approach combining the management of both previous types of clients.

Within this paper, we mainly consider the decentralized approach and discuss the technical issues related to this mode, and we demonstrate a methodology for an optimized management. Nevertheless, the simulator described hereafter is ready to accommodate the others modes of operation too.

Since there have been very few previous studies conducted on this form of management system, we are unable to make comparisons with other similar structures at present. The study of the other two approaches will form part of our future research and will not be discussed in this paper.

III. A Discrete-Event Simulator

Such a new system raises many questions, which require scientific answers. This section introduces the tool to help us providing them.

A. Multiple Questions in Need of Precise Answers

If the claimed advantages of such an open and flexible system are not merely wishful thinking, this must be substantiated by quantitative responses to the following issues.

- How many vehicles would be required to operate such a system?
- Will that number vary during the day? In which case, by how many and at what times?
- When a client meets a taxi, what criteria would be employed in deciding if he/she is accepted or refused and what would be the optimum journey for the passengers and vehicle efficiency?
- What should be done with an idle vehicle? Etc.

A model and all the resources of Optimization and Operations Research are necessary in order to predict the system performances. Obviously, we are handling a quite complex spatio-temporal decision making problem and it is almost impossible to write a mathematical model describing it with great precision. So what could the answers be and how do we know if the correct ones are being confronted?

Gaining direct experience on a "trial and error" basis would require much time to learn the system behavior and to produce an optimized structure, not to mention the client dissatisfaction and financial risk. However, a simulation is a reliable means of exploring the many aspects of a decision making problem, reproducing precisely all stochastic features and occasional unforeseen random events that periodically occur, allowing us to assess and master the potential of collective taxis at a minimal cost without the real time consequences of poor decision making. Moreover simulation is the only way to reproduce various scenarios with a single factor modified at each run. This is a fundamental property required to search for optimal policies.

B. Stochastic Discrete Event Simulations

The methods of simulation for studying many complex structures is becoming extremely diversified and used very commonly in many fields (air taxis [17], [18], Biology, [19], Physics [20], naval simulations [21], and other fields too [22].)

Types of scientific computer simulation are derived from the underlying mathematical descriptions of the problems studied:

- *numerical simulation* of differential equations that can not be solved analytically;
- *stochastic simulation* commonly used for discrete systems where events occur probabilistically and can not be described with differential equations; a special type of discrete event simulation is the *agent-based* simulation (MAS), effectively used in Ecology, Sociology, Economy, Physics, Biology and other disciplines too; indeed, many studies have been conducted on shared taxi systems using the MAS technique ([23], [24], [14], [15]).

Many open source simulation platforms ([25], [26], [27], [28], [29], [30]) and commercial ones ([31], [32], [33], [34], [35]) based on many different programming languages (C++, Java, Python etc.) are available for almost every simulation type, and they alleviate the burden of proof by their users when designing their simulation tool. Most of them treat each simulation entity as a separate thread allowing the developer to focus only on simulation specifics.

Despite all the difficulties, we chose to develop our simulation tool without the use of any particular simulation environment. Some of the implied advantages are that we can master the entire framework, having the liberty and direct access to make any necessary modifications in order to improve the tool functionalities according to the needs of the study. Consequently, we adopt a classical (not a MAS) discrete event simulation technique for learning the behavior of the "Collective Taxis" under multiple strategies.

Within this context, the system evolution is represented as a chronological sequence of the form $\{s_i, e_i, t_i\}$ where s_i is the system state at time t_i and e_i is the event happening at time t_i bringing the system to the new state s_{i+1} and so forth.

C. Modeling The System

A delicate, and also one of the most difficult, factor to establish when modelling by the Discrete Event Simulation technique is to define the set $\{(E_i, P_i)\}_{i=1,\ldots,N}$ where E_i is the set of types of events describing the system evolution, and P_i is a special procedure responsible for the treatment of any event of type E_i when activated by the event accession. Care must be taken to choose the number N of different types of events to be employed. When a very complex structure is designed, there is a high risk of producing complicated results which are difficult to understand and deal with, and consequently the value of such simulations will be considerably reduced. Conversely, if too many simplifications are taken into consideration, we shall end up with unrealistic situations and we risk coming to unreliable conclusions.

In the following, we briefly describe some types of events related to one or more of the entities involved in the system: clients, vehicles, dispatchers (in the case of a centralized or mixed mode of operation in which reservations are made through a dispatching center). For clients, we will skip some events related to clients making reservations (to save space). Clients and vehicles travel in a network and events are supposed to take place at the network nodes only.

1) Events Initiated By The Decentralized Type Of Clients:

a) Client Appears At A Node: When a client appears at the roadside seeking an available vehicle for immediate departure.

b) Client Quits The Node: When a client has found no vehicle before the waiting time limit has been reached and quits the node.

c) Client Enters The Vehicle: When a client embarks on his associated vehicle in case of a positive decision.

2) Events Prompted By Vehicles:

a) Vehicle Commences Service: At this moment the vehicle, in its initial location, becomes available to clients.

b) Vehicle Concludes Its Period Of Service: From this time onwards, the vehicle ceases to be at the disposal of clients, the event only taking place when a vehicle has an empty itinerary. If that is not the case, the vehicle will have to complete all its tasks before quitting the system.

c) Vehicle Arrives At A Node: Whenever a vehicle arrives at a node, it starts by checking if there are clients to disembark. Next, for both a mixed or centralized management, it checks if there are potential appointments with centralized types of client at this location. Then, for a mixed or decentralized management, the vehicle looks for new clients waiting at the roadside. Finally, if the vehicle is left with an empty itinerary, a station node has to be selected together with the corresponding maximum waiting period.

d) Passengers Disembark From A Vehicle: The present location of the vehicle happens to be the destination of some of its passengers and at this moment those clients alight from the vehicle.

e) Vehicle Quits Its Station: Either a vehicle responds to the request of a client, or remains stationed at a node with an empty itinerary waiting for a client until the maximum waiting time is reached, and then it must relocate to another station node, if so instructed.

f) Vehicle Ceases To Wait For Absent Clients: When a vehicle searches for its client appointments and finds one or more are absent, the vehicle checks whether and for how long it should wait. At this moment, the vehicle stops waiting for the absent client(s) and either leaves the node if its itinerary is not empty (centralized management), or it checks to examine for new candidate clients available to pick up.

g) Vehicle Completes Its Dialogue With A Decentralized Client: At this stage, it is decided whether the vehicle can accept the corresponding client or not. In the case of a positive answer, the client will embark onto the vehicle. If the client is refused, the vehicle will check if he can seek another client.

h) Vehicle Refills Its Batteries: This event concerns only systems utilizing electrical vehicles and takes place after having checked that it needs to charge its battery.

3) Events Associated With Dispatchers (Or Servers):

a) Dispatcher Starts His Service: From this time, the dispatcher is at the disposal of clients. If there are calls in the waiting queue, he selects the first one on a "first come, first served" priority basis.

b) Dispatcher Ceases His Service: If the dispatcher is not occupied with a call, he ceases his service. Otherwise, he first has to finish the task he is engaged in before ending his service.

c) Dispatcher Ends A Call: At this stage a dispatcher has given an answer to a client's request and if he has still not completed his shift, he searches for new calls waiting in the queue. He stops working when his period of duty and any call he is engaged in is complete.

D. Chain Of Events

The treatment of an event modifies the system state and it may be associated with the generation of new future events, which are then added to the event list. In Figure 1, nodes represent the event types for the decentralized management and the edges starting from each node and pointing to another one indicate the necessary creation of the event (solid line) or its possible generation (dotted line).

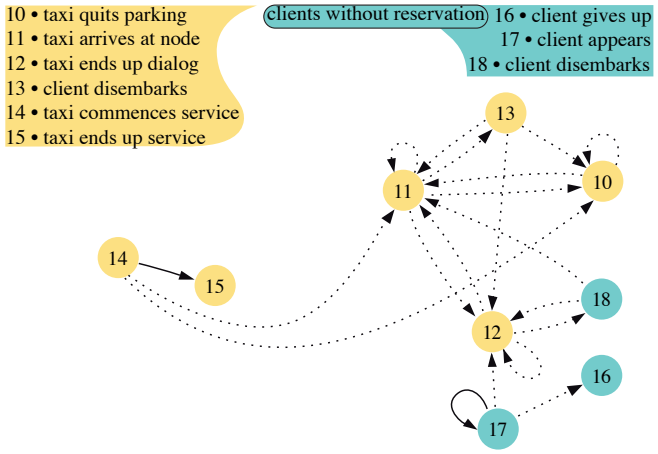

Fig. 1. Event Sequence

As an example, let us consider event type 17 corresponding to the appearance of a client at a network node. Since the client appearance is modeled by a Poisson process (that is to say, the time elapsed between two successive appearances of clients at a node is a random variable following an exponential law — see §V-A), as soon as one client appears, we have to define when the next client appearance will take place at the same node. Consequently it will be necessary to generate a new event of type 17 at this node (solid line). If the former client finds a vehicle immediately at the node, a dialogue will start between the vehicle and the client and in that case an event of type 12 will be created in order to plan the end of the dialogue. If no vehicle is found, the client decides for how long he will wait. In that case, an event of type 16 will be generated in order to put an end to the client waiting at some date in the future.

E. Decision Modeling

There is a two level set of decisions to be defined.
At the design and dimensioning stage, we have to

- model the network in which the collective taxi system will be operated (its topology, probability laws defining travel times during the entire day, ability to define congestion periods and off peak hours, client appearance and demand construction during the simulation period, definition of the client maximum waiting time at each node etc.);
- choose the operating mode (decentralized, centralized or mixed management);
- define the number of resources (number of vehicles in service, of dispatchers, etc.);
- define the service duration of each resource and its starting service time as well.

At the real time operating stage, we have to design all the control laws ruling the system. At this level, we must define decision algorithms for

- assigning vehicles to centralized types of clients;
- the acceptance or refusal of decentralized types of clients to the vehicles they meet;
- the management of the idle vehicles.

F. Simulator Design

Our goal is to conceive intelligent policies, assessing their effectiveness on a realistic virtual system closely representing a real-life structure, followed by fine tuning of all the real-time control algorithms associated with an off-line resource optimization. Evaluating the system performance for each such scenario through statistical analysis will enable us to provide an optimal configuration for any level and type of demand (offer curve).

Our initial challenge is how to construct a simulation model capable of achieving our goals. A first step in this direction consists in separating the nature of the tasks and defining the two major parts of the simulator.

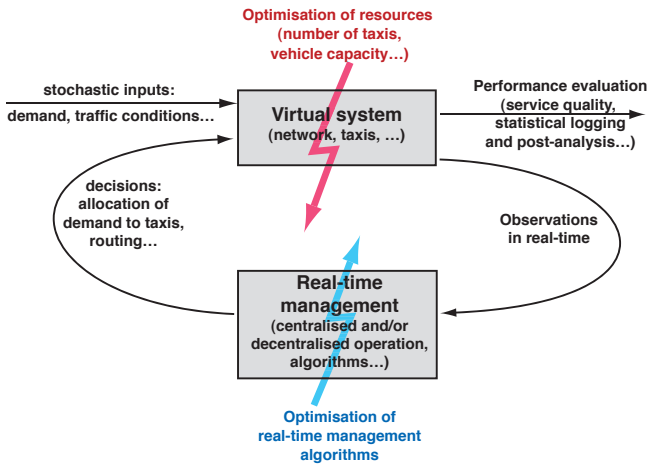

Fig. 2. The Simulator In Two Parts

The mechanical part is a virtual representation of the real system (e.g., detection of a vehicle arriving at a node). This part receives two types of data:

- *exogenous deterministic input data* related to the operational issues such as resources utilized (number of vehicles and their capacities, number of dispatchers or servers, their service durations etc.), as well as *probability laws* defining the stochastic phenomena (e.g., traffic conditions, demand, etc.);
- *decision inputs* delivered by *feedback rules* defining the real time management, which constitutes the second part of the simulator.

The mechanical part of the simulator is responsible for the evolution of the state of all agents present in the system and is in charge of interrogating the real time management when needed. Thus it detects the different states of a vehicle (arrival at a node, vehicle with empty itinerary, detection of client appointments or of clients disembarking from the vehicle, questioning of possible new clients, vehicles ceasing their service, etc.).

The real time management is responsible for all the on-line decisions required by the mechanical part. It is comprised of a set of algorithms ruling the system.

The advantages of such an architecture is, on the one hand, to allow for various experiments and comparisons of algorithms for fixed environmental data such as the demand intensity level and geometry, the topology of the network on which the system operates, the corresponding traffic conditions etc. On the other hand, for given management strategies, we intend to measure the influence of those external conditions upon the system performances.

It is well known that modeling the demand is an extremely complicated task, in particular to collect reliable data. Various methods have already been established and many new ones may be proposed in the future by specialists. Our main purpose here is to propose a consistent methodology for optimizing collective taxi performances, which would function under all potential exogenous factors. Consequently, all that information has to be modeled separately and presented to the simulation tool. The same considerations apply to traffic conditions, which may vary with the time of day and with changes of circumstance (holidays, special events, public transport strikes, road works, accidents), and to the system dimensioning concerning the number of employed resources. Meanwhile, the system control forming an individual part of the simulator, allows for a convenient evaluation of the system behavior according to various developed strategies and schemes.

Thus, for example, in order to compare two different policies concerning the vehicle management, we can proceed with two separate series of simulations, using the same simulation tool and configuration by simply modifying the real-time part related to vehicles and keeping unchanged the mechanical part and all input, including the history followed by random factors such as the client arrivals at nodes.

Similarly, if we are interested in the system behavior under some variation in resources (for example the number of the servers in the centralized or mixed managements, or the number of vehicles in service), we only modify the associated data information, everything else remaining the same.

To illustrate the respective roles of the two simulator parts, we will examine the following example. Let us consider a vehicle, which has just found a potential client waiting at the roadside. For his acceptance or refusal, the mechanical part will interrogate the real time management. This last one, by calling the appropriate decision algorithm, will reply whether the client can be accepted by the given vehicle and, in the case of a positive answer, the optimal vehicle itinerary will also be provided.

In this kind of decision, if one attempts to keep detours for passengers, with respect to their most direct itinerary, at a very low level, then this quality of service for the existing passengers is likely to penalize new candidates since rejection will be more frequent, and consequently waiting times will be extended, leading to a greater client abandonment rate. This consideration shows that such a decision algorithm must include parameters which will be fine tuned by repeated experiments. This is a first level of optimization regarding the on-line management.

The second optimization level involves the mechanical part, and more specifically the system dimensioning (number of vehicles and/or dispatchers in service according to the system management, their duration of service, the vehicle capacity, etc.).

G. Simulator Features And Options

1) Simulation Time: The simulation keeps track of the present simulation progress, which is the time of the current processed event. The treatment of each event is instantaneous and the simulation time advances as we pass from one event to another. For our studies, the time unit considered is one second.

2) Event Stack: The simulation maintains a pending event set comprised of all the events that are not yet processed. More precisely, the event list is organized as a priority queue, sorted by the event time. Thus the event list is of the form $\{(e_1, t_1), (e_2, t_2), (e_3, t_3), \dots\}$ where e_i is the event occurring at time t_i with $t_i \leq t_{i+1}$. In our system the event list is not entirely sorted. It suffices to have the earlier event at the head of the list whilst the rest are not necessarily ordered.

3) Simulation Duration: Before any implementation, the simulation duration must be specified. This value expresses the real time period during which we wish to explore the system behavior.

For the experiments reported later on in this paper, we have observed that simulations of 8 hours (of real time) with steady data (fixed probability laws of random inputs, fixed resources) yield statistical estimators with a very low variance. Therefore, we adopted this value for the simulation duration.

The simulator can be used in various ways according to the needs of each particular study. We describe these options now.

4) Specifying The Initial Condition Of The System: Before any simulation run, we have the opportunity to specify precisely the initial state of the system. We may want to start a "new" simulation, which implies that the initial system state is empty and consequently the event list contains only events concerning system initializations. More precisely, all the system resources must be put in service (vehicles have to be positioned at nodes, servers or dispatchers should be at clients'disposal, etc.) and clients should start to appear in order that the interaction between these two types of agents (clients and vehicles) will start taking place.

Alternatively, it is possible to define a desired initial system state. We may need to continue a previous simulation run if we consider that its duration was not long enough to form valid conclusions. In that case, the initial state of the system is given and it is the final one of the previous simulation. The event list will contain all the events that were not processed during the previous simulation (since their times were greater than the allocated simulation time).

5) Specifying The Clients Utilized: One possibility for client appearance is to generate new clients dynamically during the implementation according to the demand probabilistic model.

But we can also employ previously generated clients (during another run) subject to the duration of the new simulation not exceeding the duration of the particular run from which we wish to re-use the clients. It turns out that this is an indispensable option for the optimization of the system performances when we are interested in analyzing how the same client history behaves under various policies or even simply under different system conditions.

6) Simulation Loop: To summarize, a simulation run requires an initialization phase which consists in

- defining the simulation duration;
- specifying the nature of the simulation (new simulation or continuation of a previous one) as well as the type of clients utilized (generation of new ones or using a record);
- setting the clock to the starting time (this is an input in case of a new simulation, or it is the time at which the previously completed one ended, detected by the simulator engine);
- fixing resource parameters and initializing the system state variables (number of resources utilized, the duration of their services, vehicle positioning, etc.);
- scheduling the event list (with the necessary bootstrap events if a new simulation is desired, or by collecting the previously unresolved events remaining in the event list).

Then the simulation proceeds by

- handling the first event of the list; by the end of its treatment, it will be removed from the list;
- updating the clock to the time of the next event in the list…
- or stopping the simulation if the next event takes place after the specified terminal date.

H. Recording And Analyzing Results

The purpose of simulations is to accurately reproduce the system behavior according to any possible scenario. The original architecture of the simulator, separating the nature of tasks (controls from mechanical schemes) and the simulation technique employed (independent of threads and techniques utilized on multi agent simulations) within our study, requires us to face different types of problems from those listed in many other papers such as [23], [24], [14]. The evaluation of the performances of the applied strategy and all the numerical results and conclusions are not forming part of the simulator. On the contrary, once more, we prefer to separate these tasks, which take place during a following stage. More precisely the simulator registers *all* treated events, which are then stored in a database for later exploitation. This represents an enormous volume of information provided by each run, but any question that may be raised a posteriori can in principle be answered as long as an exhaustive record of all events has been kept. Exploiting that information is a complicated work, which is one of the reasons why we prefer to disconnect it from the main simulation run.

For this second phase of analysis, several scripts have been developed with two main objectives in mind.

- Either one is willing to perform a microscopic analysis of a specific run by tracking particular sequences of events or pinpointing seemingly abnormal functioning: this is in particular the case at the debugging stage in the

development of the program, or when wrong decision rules have been implemented.
- Or one is interested in a macroscopic, rather statistical, analysis of one, or possibly a series of, run(s) in order to compare several scenarios

Examples of both types of analysis will be given at §V and VI, respectively. A microscopic analysis requires interactive scripts whereas the macroscopic analysis is rather a batch process.

I. Programming Language

For the collective vehicles simulator, we chose to develop the program in "Python", which is a dynamic object-oriented language offering strong support for integration with other languages (C, C++ etc.) and software tools, encouraging an easily maintained, clear and high quality development offering multi-platform versatility (Mac OS X, Linux/Unix, Windows). Thus we are employing programming techniques such as encapsulation, inheritance, modularity to design all the particularly complex applications to be managed and implementing all the involved objects and their interactions within a relatively simple environment. These features become especially useful since the goal is to be able to reuse code for various approaches to collective taxis.

IV. Client Acceptance Algorithm (Decentralized Mode)

As previously explained in §III-F, every time there is a decision to be taken, the mechanical part asks the real time management for the necessary answer. Hereafter we focus on the decentralized approach in which the main real time decisions concern the question of what to do with empty vehicles and when to accept clients on board of taxis.

Among all the decisions composing the real time management, an extremely important part is the one referred to as the client acceptance by vehicles. In this paper, we shall consider a simple algorithm with a parameter to be tuned with the help of the simulator. However, there are various other possible algorithms for which the simulator can also prove useful to optimize them and assess their performances.

Every time there is a dialogue between a client and a vehicle, there is a binary decision to be taken concerning the client acceptance by the given vehicle. This decision is closely related to a second one concerning the possible new itinerary of the vehicle. The aim of the decision algorithm is to decide whether the new client should be accepted and, if so, to provide the new itinerary.

A. Notations

- Node n_0 is the present position of the taxi, t_0 is the present time, and $L = \{n_1, n_2, \ldots, n_m\}$ is the *sorted list* of nodes of its itinerary.
- We are searching for an optimal order for list $\ell = L \cup \{n_c^d\}$ where n_c^d is the destination of a candidate the taxi just met (M is the number of distinct nodes in ℓ).
- $\delta(n_i^o, n_i^d)$ is the duration of the direct travel (shortest path computed using average travel times on all arcs of the network) from origin n_i^o to destination n_i^d of client i (matrix δ is precomputed as explained later on).
- We denote by $\ell_1 = \{x(1), \ldots, x(|\ell|)\}$ any considered tour to visit *all* nodes in ℓ. In addition, $x(0) = n_0$.
- $p\big(x(k)\big)$ is the number of passengers disembarking at node $x(k)$.
- $u(k), k = 0, \ldots, M-1$, is the chosen next visited node when the vehicle is at node $x(k)$; consequently $x(k+1) = u(k)$.
- $t(k)$ is the predicted arrival time at node $x(k)$, which is easy to compute using matrix δ: $t(0) = t_0$ and $t(k+1) = t(k) + \delta(x(k), u(k))$.
- $\mathcal{E}(k)$ is the set of all visited nodes at step k; we consider that $\mathcal{E}(0) = \emptyset$; obviously $\mathcal{E}(k+1) = \mathcal{E}(k) \cup \{u(k)\}$.
- s is the *detour threshold* with respect to direct travel: *this parameter is to be tuned by repeated simulations* as discussed later on.
- $t^{\lim}(j)$ is the deadline for arrival at node $n_j \in \ell$:

$$t^{\lim}(j) = \begin{cases} t_0 + s \times \delta(n_0, n_c^d) \\ \qquad \text{if } n_j = n_c^d \text{ and } n_c^d \notin L, \\ \max\Big(\min\limits_{i \in d^{-1}(j)} \big(t_i^o + s \times \delta(n_i^o, n_i^d)\big), t_j^p\Big) \\ \qquad \text{if } n_j \in L. \end{cases} \quad (1)$$

The idea behind this formula is that, for each passenger, the detour threshold s should not be exceeded, as a proportion of the likely duration of his direct travel (the one he would have made if he had chosen a classical taxi or used his own vehicle). However it may be already impossible to satisfy that constraint due to past vehicle delays for doing various operations at nodes, stochastic travel times, etc. In that case, the retained deadline associated with node j will be its predicted arrival time t_j^p *before* the new candidate is accepted (that is, t_j^p is computed using the *original* sorted list L).

B. Problem Statement

The purpose is to minimise

$$\sum_{k=1}^{M} p\big(x(k)\big)\, t(k)$$

by choosing ℓ_1, that is, an order for the unsorted list ℓ, subject to the constraints:

$$x(k+1) = u(k),\ k = 0, \ldots, M-1,\ x(0) = n_0,$$
$$\mathcal{E}(k+1) = \mathcal{E}(k) \cup \big\{u(k)\big\},\ k = 0, \ldots, M-1,\ \mathcal{E}(0) = \emptyset,$$
$$t(k+1) = t(k) + \delta(x(k), u(k)), k = 0, \ldots, M-1, t(0) = t_0,$$
$$u(k) \notin \mathcal{E}(k),\ k = 0, \ldots, M-1,$$
$$t(k) \le t^{\lim}(x(k)),\ k = 1, \ldots, M,$$

where $t^{\lim}$ can be precomputed independently of the problem solution using (1).

C. Resolution

Such a problem can be solved by Dynamic Programming [36] using the state vector $(x, \mathcal{E}, t)$. However, when vehicles have a moderate capacity (say, 5 passengers maximum), the

enumeration of all possible orders ($5! = 120$ at most) is feasible. For every order, we have to recursively compute the $t(k)$ (and the cost function simultaneously) and check whether they exceed the deadline, and consequently stop exploring that order as soon as the deadline constraint is violated.

Finally, we keep the best feasible order according to the cost function. In case of greater vehicle capacities, another (suboptimal) alternative is to try inserting the candidate node at each possible position of the given itinerary without changing the order provided by the original list L.

V. Implementing The Simulator And Scanning The Results

We now proceed to implementation and explain how results can be analyzed. For the time being, our study has been limited to the decentralized management although the simulator is ready for the other two operation modes. The main difficulty is to design real time decision algorithms, a task which is more difficult and which requires further research in the case of the centralized and mixed management.

We start by showing the data utilized in this study. At this stage of our research, for reasons that we briefly discuss hereafter, we had no access to real data, hence the data used for our experiments have been built up with the objective of looking realistic enough for a city like Paris (except for the network depicted in Figure 3 whose description is somewhat sketchy). Indeed, our main goal for the time being is to establish a methodology to evaluate such a collective taxi system. Different sets of data will provide different numerical results, but they should not affect the method of producing results and exploiting them.

Throughout the world, legislation governing the use of taxis is very restrictive, especially regarding the decentralized management considered here, in which clients are picked up in the streets. This has the unfortunate effect of leaving little room for intelligent systems with potentially successful results and the implied benefits for customers in terms of cost, service quality, and environmental advantages (parking, congestion, fuel consumption and its impact on pollution). This is probably one of the main reasons why we had to launch this initiative without waiting for partners ready to consider a real life implementation, hoping that this step would contribute to remove some of the present locks in the future.

A. Data Utilized

1) Network: The area in which the system operates (a city possibly including the suburbs) is represented as a set of nodes and edges with their associated properties. All events take place at nodes while vehicles travel on edges. With each edge, there is an associated random travel time following a shifted log-normal distribution (the shift ensures a minimum positive trip duration). These laws may vary during the simulation period to represent changing traffic conditions.

In general decisions are based on mean values of those quantities for future events, but, during simulation, the actual movements of vehicles are based on pseudo-random values following the given laws. For speeding the simulation up, a

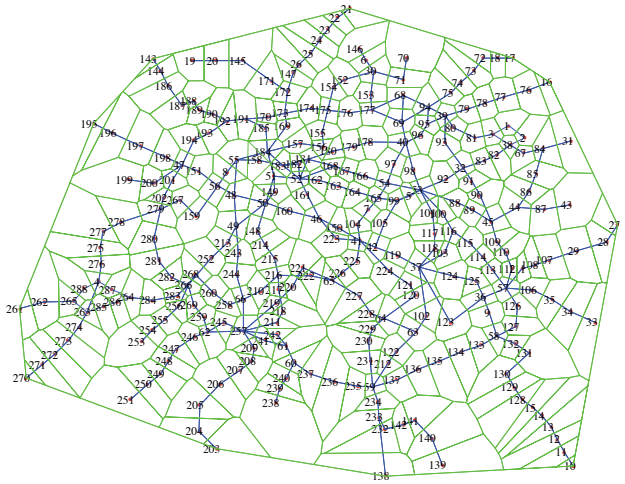

Fig. 3. The Network

matrix δ of the durations of shortest paths for all pairs of nodes is calculated in advance, using the average traveling times of arcs. Another matrix S, calculated simultaneously and also stored, provides the route for following the shortest path (namely, if $k = S(i, j)$, then k is the next node to go to when the vehicle is at node i and is willing to join node j by the shortest path; at k, it will go to $S(k, j)$ and so forth). The computation of matrices is performed using a program in C provided by S. Gaubert [37] and based on Howard's algorithm [38]. This program can be interfaced with the script language of Scilab software [39] and its network analysis toolbox Metanet [40] which we used to manipulate our network data.

The network we have used is comprised of 288 nodes and 674 edges (indeed, it is inspired by the Paris metro plan — see Fig. 3).

2) Demand: Clients arrive at nodes according to a Poisson process: the elapsed time between successive arrivals at node i follows an exponential law of parameter λ_i.

Moreover every such client receives a randomly chosen destination node. This choice follows the conditional probability law of destinations, given the origin, the so-called O-D (origin-destination) matrix.

The parameters of Poisson processes at nodes determine the demand intensity, and this may vary within the time (as also will the O-D matrix) to represent fluctuations of the demand in volume and geometry.

For the next reported results, the average number of clients per hour on the whole network is about 15,400, whereas the flow from origin to destination is mostly centripetal (clients are moving mostly from the suburbs towards the centre of the town). But we are able to modify those characteristics at will and to reoptimize the system for any given demand as demonstrated at §VI.

3) Vehicles: A number of vehicles operate at each period of the day. Each vehicle is characterized by its seating capacity and service time. For a vehicle with an empty itinerary, a decision must be made indicating what must be done with it (either it will park for some limited time at its present position,

or it will be directed toward another node for parking, unless it meets new clients in the meanwhile).

For the following run, we initially positioned 13 vehicles per node (3,744 overall), but this distribution changes rapidly as the the simulation progresses. This figure has not been chosen at random of course, but after some experiments. It must be adapted to the demand level and/or geometry as explained at §VI-B.

The capacity of each vehicle is 5 passengers and we will study its influence at §VI-C.

4) Detours: The detour threshold parameter s was introduced at §IV for limiting the detours borne by passengers with respect to the expected duration of their direct travel form origin to destination. At §VI, we will study the influence of this parameter upon performances and see how it can be tuned to achieve reasonable trade-offs between conflicting indicators of the quality of service offered to customers. In the following experiment, it is set to 1.9 (again not totally at random!).

5) Durations: We allocate 30 seconds for each dialogue between a candidate client and a vehicle and 10 seconds for each embarking or disembarking of passenger.

The client maximal waiting time at his origin node is 10 minutes (that is, if no vehicle was able to accept it after 10 minutes, the customer gives up and this will be reflected in the abandonment rate reported below).

The maximal parking duration of an empty vehicle is 15 minutes: after this time is exceeded and no client showed up, a decision must be made about staying at the same node for a new parking period or departing toward another node.

Those duration values are not necessarily fixed but may be modeled as random variables to represent for example more or less patients customers.

B. Statistical Verifications

Before analyzing the results of a simulation run, we can check if the random data generated during the implementation have conformed to the corresponding probability laws. This mostly concerns the demand and also travel times.

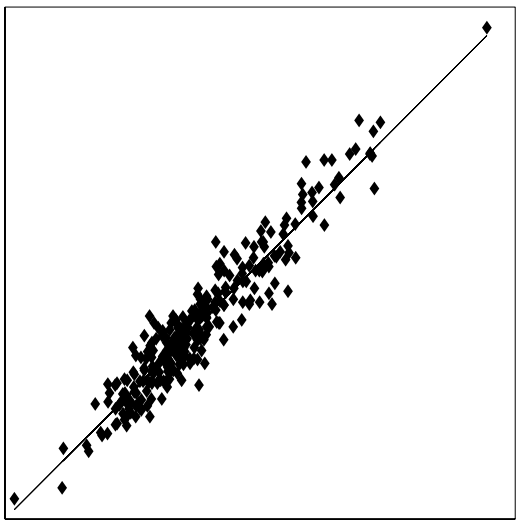

Fig. 4. Checking The Demand Intensity

Figure 4 represents the statistical verification of parameter λ_i corresponding to the intensity of client appearance at node i (see §V-A). The x-axis represents the theoretical value of λ_i and the y-axis represents its estimated value, for $i = 1, \ldots, 288$. Therefore, the plot is made up of 288 dots, which should ideally be aligned along the first diagonal.

The same visual technique can be used to compare theoretical and estimated values of the frequency of each destination given the origin (O-D matrix), and the average travel time of vehicles through each edge of the graph.

C. Microscopic Analysis

In the sequel of this section, we show the wealth of information that a simulation can provide.

1) Network Results: Figure 5 shows the number of visits of nodes by vehicles over the whole simulation run (the x-axis corresponds to the node numbers). The same kind of plot can be obtained for how many times vehicles crossed each edge of the graph. This is a quick way of discovering

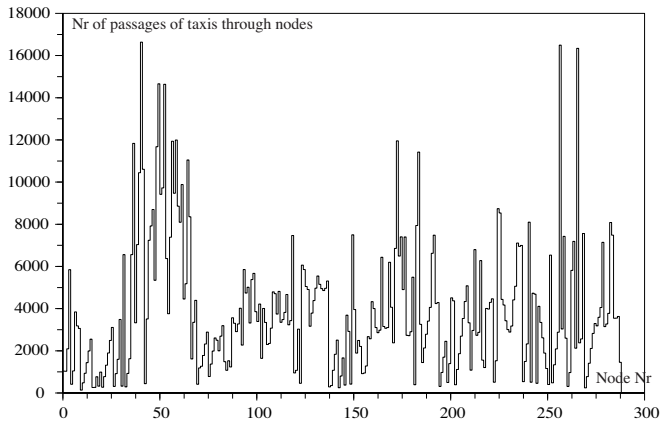

Fig. 5. Visits Of Nodes By Vehicles

parts of the network which may not be well served by the decentralized management policy adopted here and under the prevailing conditions (in particular the geometry of demand).

2) Client Waiting Time And Abandonment Rate: Figure 6 represents the mean client waiting time ($\pm$ the standard deviation) at each network node and for the whole network (horizontal lines).

In order to detect critical network nodes, we may also examine the abandonment rate: as explained earlier, clients not served after 10 minutes give up and we measure the proportion of such clients at each node. The average client abandonment rate for the whole network is 1.33% as shown in Figure 7 by the horizontal line, but the node-per-node abandonment rate displayed in the same figure reveals that it have reached a very high value close to 45% at Node 10. In order to locate this node, it is easier to consider Figure 8 which is a colored map in which each cell is a Voronoi cell relative to a node of the network (a Voronoi cell is the set of points in the area which are closer to that node — called the "centroid" of the cell — than to any other node of the net): the shade of a cell is proportional to the abandonment rate of its centroid, and darker shades represent higher values of this rate. Node 10

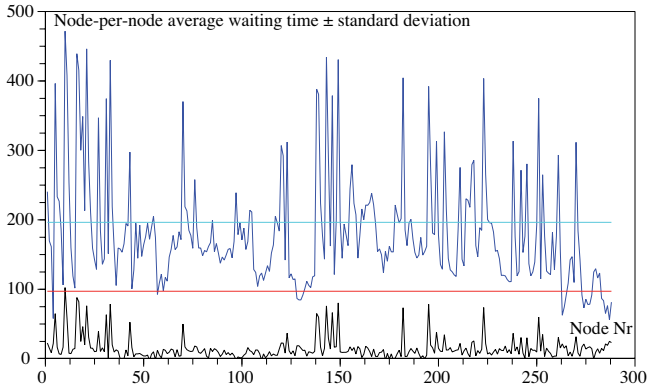

Fig. 6. Client Waiting Time At Each Node

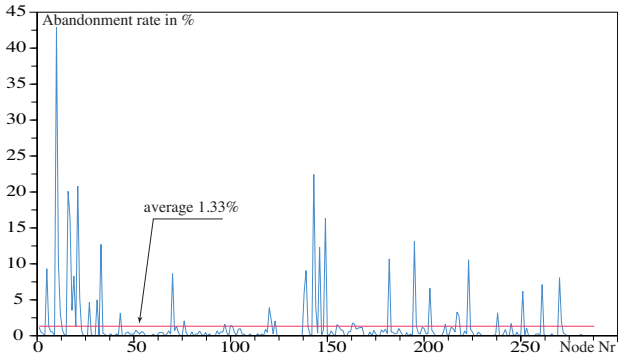

Fig. 7. Client Abandonment Rate

is located at an extreme South-East position in the area. We also notice other "bad" nodes mostly located at the periphery of the city. This observation is not surprising, given that the demand is mostly centripetal, which tends to concentrate taxis in the the city center.

However, some dark shades also appear in the city center, the most noticeable one corresponding to node 149. During our analysis session using SCILAB scripts, we can interactively ask for a plot of the evolution of the client queue length at

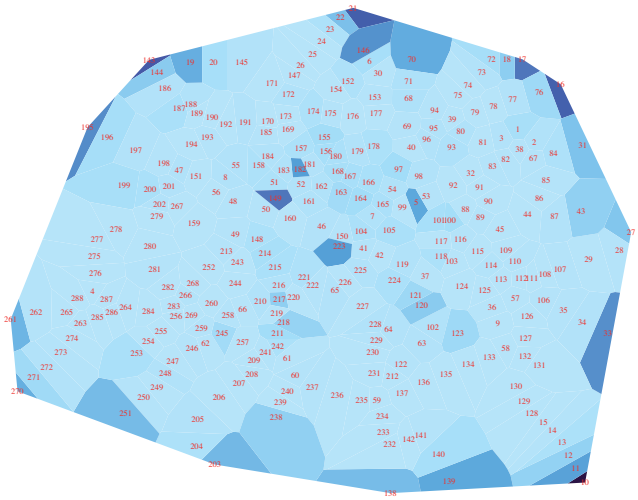

Fig. 8. Map Of Client Abandonment Rate

this node over the whole simulation, and this plot shows that this queue length quite often reaches rather high values. We may also observe the rather small number of visits of this node by vehicles in Figure 5. All those observations lead us to proceed to a deeper analysis for this particular node. Figure 9 shows a small portion of the network around Node 149 with the frequency of visit of the nodes by the vehicles (average number of passages per minute, in parentheses). All edges between nodes can be used in both directions (indeed, in the internal representation of the graph, all edges are directed, that is one-way, but for clarity of the drawing, only one line is drawn here for both directions). Obviously, from Node 50

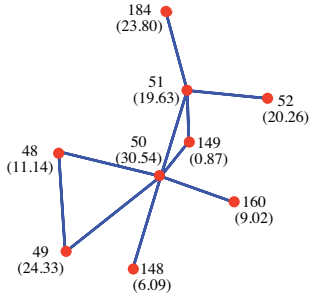

Fig. 9. Zoom Around Node 149 (figures in parenthesis are the frequency of visits by taxis in nr/min)

to Node 51 (back and forth), vehicles go mostly straight and bypass Node 149 by using a parallel route. This particular configuration of the network explains the bad situation at Node 149.

This example shows the ability of our simulation tool to quickly detect flaws and misconceptions of the system.

3) Taxi Activity: During this simulation run, taxis carried from 2.63 to 5.13 clients/hour (these are the extreme values observed over all taxis in service) with an average of 3.77 clients/hour. The average number of passengers per vehicle is 2.16 (individual values vary from 1.29 to 3.15). Recall that the capacity of taxis in this simulation is 5 passengers, that is, in the average, vehicles are not crowded.

Figure 10 shows the percentage of time spent by vehicles with n passengers on board. On request, such histograms can be produced individually for particular taxis.

We are now interested in examining how busy taxis are. Figure 11 shows the percentage of time spent by vehicles on travelling (largest area), doing various operations at nodes (examining new candidate clients, embarking or disembarking passengers — medium area) and finally being idle (smallest area representing only 3% of the total time of service).

4) Quality of Service Provided: We explore now an indicator deserving a particular attention since it is an important ingredient of the *quality of service* provided by the collective taxi system.

We define the "*total detour ratio*" as the ratio of the "actual duration of client's travel plus the initial client's waiting time" over "the duration of client's direct travel" (the latter being evaluated by the shortest path from origin to destination using the average travel times on edges).

Remember that, for the considered run, the parameter s introduced at §IV (see in particular (1)) has been set to 1.9.

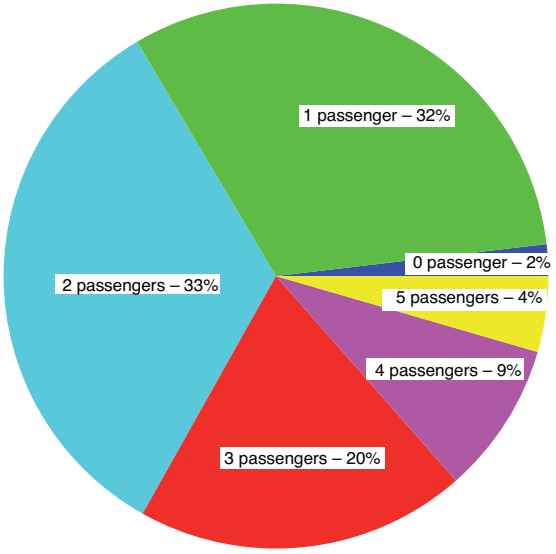

Fig. 10. Percentage Of Time With n Passengers On Board

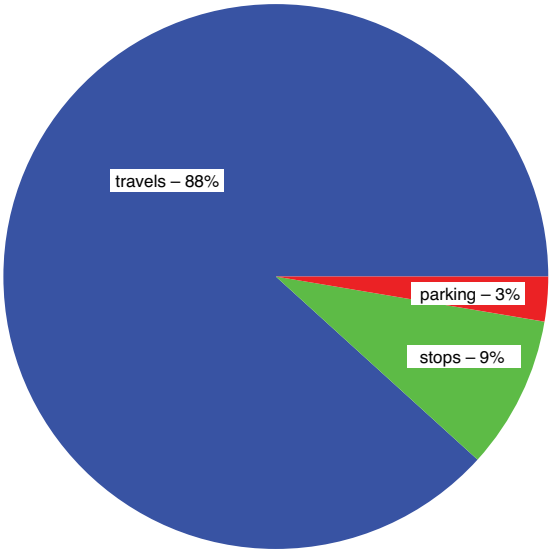

Fig. 11. How Busy Taxis Are

This parameter was introduced in the client acceptance algorithm as a means to moderate, if not absolutely constrain, the "simple detour ratio", which differs from the "total detour ratio" introduced here by the fact that the initial waiting time is not included at the numerator of the ratio (therefore, the total detour ratio is greater than the simple detour ratio). Indeed, if s is decreased, the resulting simple detour ratio will likely decrease too, but, as explained earlier, the initial waiting time is likely to increase (because of a higher probability for candidates to be rejected). Hence the total detour ratio is a nice indicator of the quality of service since it incorporates two conflicting quantities which must be balanced.

Figure 12 presents the histogram of the total detour ratio for all the served clients during the simulation. It may seem strange that a small part of the histogram lies below the value 1. However, remember that the denominator of the ratio defining the total detour ratio uses *average* travel times

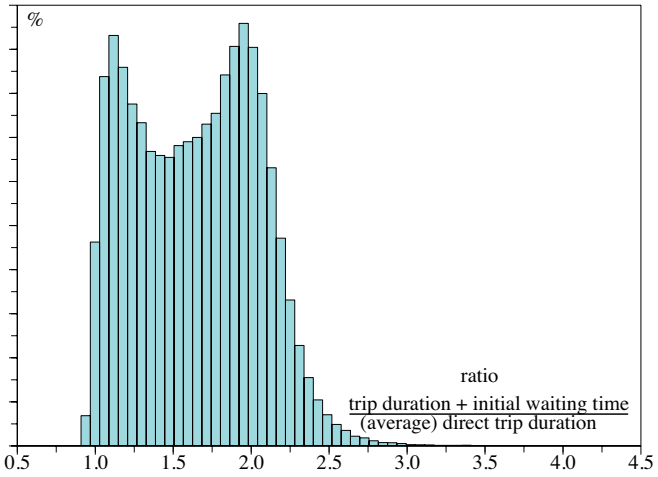

Fig. 12. Total Detour Histogram

on edges, whereas the numerator involves *actual realizations* of those travel times, which may sometimes fall under the average. For the same reason, and also because when s is used in the algorithm, it is applied to trip durations which are partly realized and partly predicted, it should not be surprising either that a part of the histogram lies beyond the value 1.9 adopted for s (plus again the fact that the total detour ratio incorporates the initial waiting time, while s limits only the simple detour ratio).

Nevertheless, it turns out that the average value of the total detour ratio is 1.64 (notably less than the value of s) with a standard deviation of 0.4.

Finally, one may have noticed the two-hump shape of this histogram, a permanent observation in all our experiments (which applies also to the histogram of the simple detour ratio). This suggests the existence of two (or perhaps more) categories of clients. Several explanations have been imagined and tested by additional investigations. We refer to [41] for such a discussion. One (albeit partial) explanation (proposed by F. Meunier) has to do with the number of passengers already on board when a candidate is accepted by a taxi. Indeed, if the histogram is split into separate histograms for clients accepted in empty taxis, clients accepted when one passenger is already on board, etc., it turns out that the left-hand side hump is predominant for clients embarking in empty taxis, and then this hump tends to disappears compared to the right-hand side one when the number of passengers already on board increases.

VI. Tuning Parameters

So far, we have shown some of the information that can be extracted by the detailed analysis of single run of the simulation program using interactive Scilab scripts. Among the parameter which must be set off-line, let us mention

- the detour threshold s introduced by the client acceptance algorithm;
- the number n of vehicles in service;
- the capacity (maximum number of passengers) of these taxis.

The purpose of this section is to work out a methodology to analyze a series of simulation runs in which one or several of these parameters take a range of values, with the purpose of choosing "the best values". Indeed, given the huge amount of indicators of the quality of service offered to clients, and also of the cost of operation, that can be obtained from each run, the real problem is to select a small number of truly relevant indicators on which to base our choice. As we shall see, the same methodology can also be used to assess the impact of other external factors that we do not directly master, such as the demand, in intensity and geometry.

A. Relevant Indicators And Methodology

Initial waiting times of clients, queue lengths at nodes, detours with respect to direct trip, etc., are some of the numerous indicators of the quality of service offered to clients. Some of them are strongly correlated with each other (we have checked this for waiting times and queue lengths at nodes for example) so that it is sufficient to monitor one of them. Some generally vary in opposite directions under the influence of control parameters: as we observed it already, decreasing the value of s will likely also make the average detours decrease but waiting time increase. In that case, the notion of "total detour" introduced earlier is a compact way of taking both effects into account in a single indicator.

However, notice that such indicators are only relevant for clients that finally became passengers, that is, who finally succeeded to get on board of a taxi. Those who gave up after their maximum waiting time was elapsed are definitely lost: they must be specifically taken into account through the abandonment rate.

Regarding the number n of taxis in service, increasing this value will certainly impact all indicators of the quality of service positively from the point of view of clients. But the price to be paid will be a reduced commercial activity of each vehicle, meaning that fares should be raised in order to ensure the profitability of the system. By the way, it is perhaps time to say that, in the same way as we made no assumptions about the feedback between the quality/fare ratio offered by the system and the demand attracted by this system (demand is an independent input of the simulator), we did not either postulate any particular fare level or tariff structure (fixed, proportional to the direct trip length, etc.). We limit ourselves to extract from simulations the relevant information allowing to evaluate the turnover and associated operation cost for taxis. For example, taking the assumption of the simplest tariff structure, namely, a fixed fare for any passenger, we can measure the numbr of customers served by each taxi during a simulation. With a fare proportional to the direct travel time, a more relevant information would be the total of such direct travel times for all customers transported by each taxi.

Finally, the following three indicators will be especially monitored:

- x: the average client abandonment rate throughout the network;
- y: *minus* the average number of transported customers per vehicle (the *minus* sign will be justified hereafter);
- z: the average total detour ratio of all served passengers: recall that z incorporates two statistical indicators, the *client initial waiting time* and *detours* born by passengers already arrived at their destination.

Indicators x and z improve (decrease) when the number of vehicles in service increases, whereas the average number of clients served per taxi is likely to worsen (that is, also to decrease). Therefore, by choosing "minus" this number (as y is defined above), for all three indicators, *better* now means *smaller*.

In the following, we start by keeping the capacity of taxis constant throughout all runs of a series, but let the threshold detour parameter s and the number n of vehicles in service vary. Then, each run in the series corresponds to particular values given to (s, n), and this run produces a point in a 3D space with coordinates (x, y, z). This cloud of points depending on two degrees of freedom in a 3D space should draw a surface.

Amongst these points, we need to pay attention only to the *non dominated* ones (points for which there are no any other point which is better according to the three coordinates simultaneously — Pareto optimality). We seek values of the parameters (s, n) for which the three selected indicators become *as small as possible*, but of course no point will dominate all other points. We have to make a "reasonable" trade-off amongst the three indicators by choosing some point lying on the surface, which in turn will determine the value of (s, n) to adopt.

Nevertheless, when we compare two situations in which, for example, the demand differs, each demand assumption will provide such a surface and, if one surface is above the other one, it means that, even before choosing a suitable trade-off, we can say that the demands can be ranked as more or less favorable.

In what follows, we illustrate this methodolgy by studying the impact of some factors such as the demand and the capacity of taxis. We start with the influence of demand geometry (mostly characterized by the O-D matrix) but the reader may refer to [41] for similar results corresponding to varying the intensity of demand (characterized by the parameters λ_i introduced at §V-A2).

B. Influence Of The Demand Geometry

We have constructed three different geometries of demand :

- the *centripetal* one where clients move mostly from the periphery towards the city center;
- the *centrifugal* geometry in which the outskirts are more attractive;
- the *balanced* one in which each node emits the same number of clients that it attracts.

Those scenarios are tuned to correspond to the same demand intensity (average number of clients appearing at all nodes per time unit). We refer the reader to [41] to see how this is mathematically done by playing with the O-D matrix M and the vector λ of Poisson process parameters. It suffices to say here that, considering the centripetal and centrifugal demands for example, they intuitively correspond to the movements of

people going from home to work in the morning and returning back home in the evening.

We proceed to two series of simulations when varying values of s and n for the centrifugal and centripetal demand scenarios. In Figure 13, the lower surface corresponds to the centripetal demand and the upper one to the centrifugal. Therefore, one can assert that the centripetal geometry is more favorable.

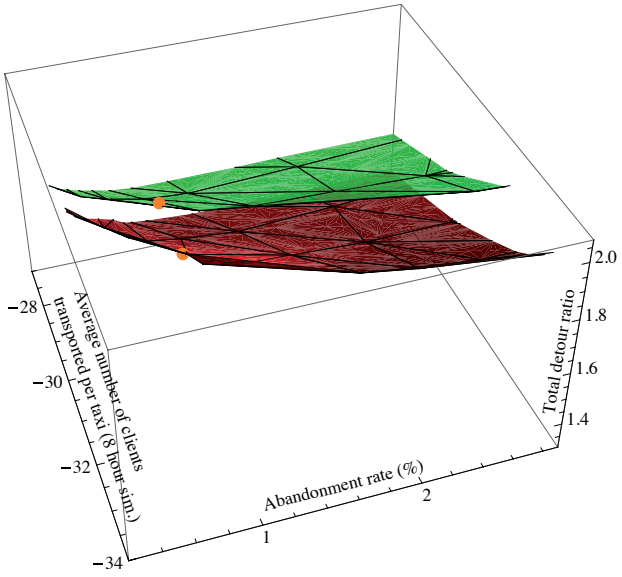

Fig. 13. Comparison Of Centripetal And Centrifugal Demands-3D

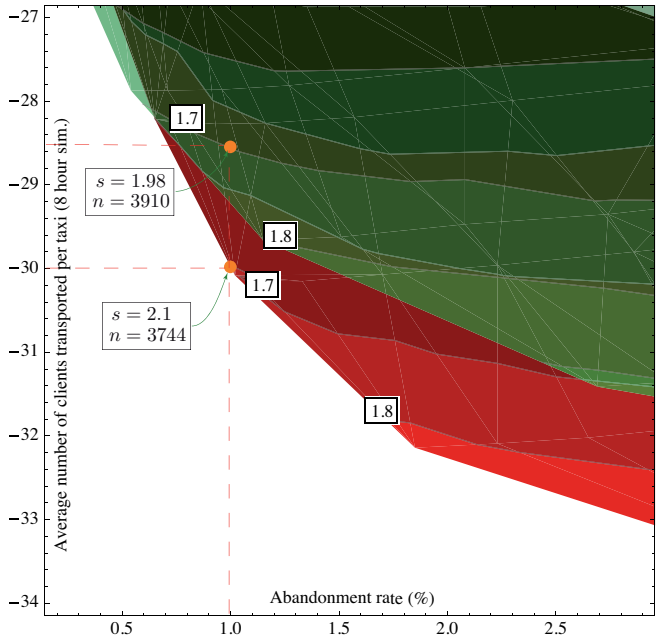

Fig. 14. Comparison Of Centripetal And Centrifugal Demands-2D

In order to quantify this advantage, for each of the two scenarios, we choose a particular point on the corresponding surface in such a way that two of the three coordinates of those points are equal, and we look at difference over the third one. For this operation, it is convenient to use the 2D representation of the surfaces in the (x, y) plane while the z coordinate is represented by its level curves. The level curves are indicated in square boxes in Figure 14. The points chosen on each surface are located on the level curve 1.7 (that is, both achieve an average total detour ratio of 1.7) and they also both achieve a global abandonment rate (x coordinate) of 1%. However, the corresponding average number of transported clients per vehicle $-y$ is greater with the centripetal geometry (about 30 clients for simulations corresponding to 8 hours of real time) than with the centrifugal (only about 28.5). This is consistent with the fact that more vehicles are needed with the centrifugal demand to achieve those performances (3,910 versus 3,744). Finally, a different value of s is also needed: 1.98 versus 2.1.

We can conclude that the centripetal geometry of demand in a network having a topology inspired by the Paris metro plan, tending to accumulate vehicles towards the city center, is more favorable than the centrifugal one, which tends to disperse taxis toward the suburbs. Of course, other topologies may lead to different conclusions.

C. Varying The Vehicle Capacity

In this section, we are interested in the following question: what is the optimum seating capacity and corresponding number of taxis for maximum efficiency? Just as a preliminary study in this direction, we compare series of simulations using either taxis with capacity 5 or 7.

As mentioned at §IV-C, when passing from capacity 5 to 7, it is no longer possible to use the exhaustive enumeration of all possible orders to search for the itinerary which solves the optimization problem of §IV-B: this is too computationally expensive. Therefore, we have used the suboptimal strategy which consists in trying to insert the new candidate at all possible positions in the existing itinerary of the taxi. We mention that experiments conducted with taxis of capacity 5 in order to compare the exhaustive enumeration and this suboptimal solution led to the following conclusion: out of the cases when there are 0 or 1 passenger already on board — since, in that case, there is actually no difference at all between the two strategies —, we observed that the solutions delivered by the two algorithms were different in only 0.5% of the cases. Of course, we do not claim that this conclusion is still valid with capacity 7.

Figure 15 represents the surfaces corresponding to simulations with 5 (upper surface) and 7 passenger seats (lower surface). In this 3D figure, the so-called upper surface indeed crosses the lower surface. Consequently along that curve of intersection, the two systems present the same performances.

Let us now have a closer look at the 2D-representation of Figure 16. As previously, the level curves display the value of z (total detour ratio, now indicated in circles). Consider a particular point at the intersection of the level curves corresponding to $z = 1.8$ (average total detour ratio): this point is obtained with about the same number of vehicles (4,104 versus 4,105 taxis, for capacities 5 and 7, respectively) and for slightly different values of s (2,02 versus 1,96). In both cases, the abandonment rate is $x = 1.85\%$, and the number of

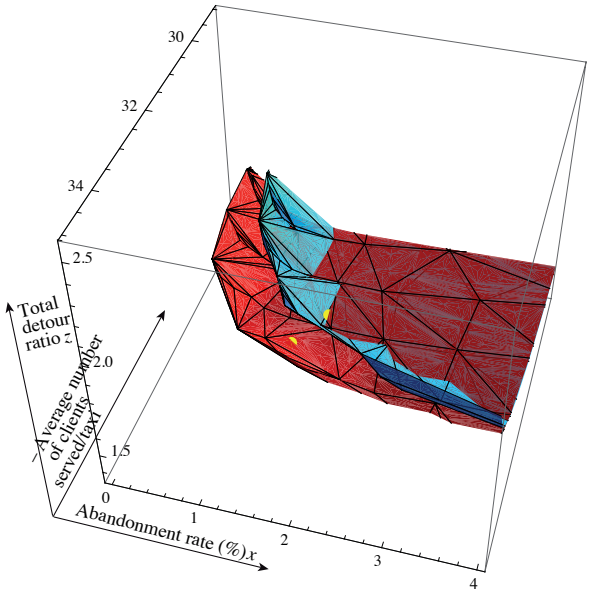

Fig. 15. Comparison Of 5 And 7 Passengers Capacities-3D

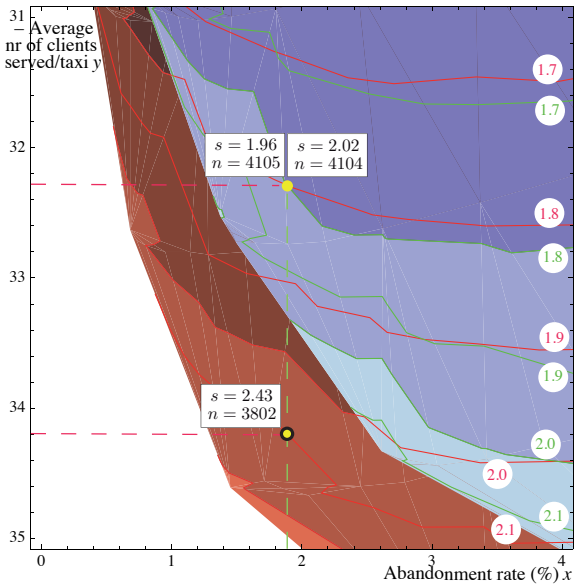

Fig. 16. Comparison Of 5 And 7 Passengers Capacities-2D

transported passengers per vehicle $-y$ is approximately equal to 32.25. Therefore, for such an operating point, it is apparent that the greater vehicle capacity does not bring any significant contribution, that is, the additional capacity is not really used.

If we consider now the alternative performance point of Figure 16 (circled in black), which is accessible only by the system employing 7 passenger vehicles, we observe that the number of transported clients per vehicle does increase to 34.2, while preserving the same value of the client abandonment rate $x = 1.8$ as previously. However, the average total detour ratio jumps to 2.1 i,nstaed of 1.8 previously. We conclude that this configuration allows the transportation of more clients over the 8 hour simulation(therefore costs would be lower), but at the expense of providing a lower quality of service.

In conclusion, the smaller vehicle capacity is adequate to gear the system toward performances that are more "taxi oriented" (although already "collective"), whereas the larger capacity turns the system more toward a collective transportation mode.

VII. Conclusion

The Collective Taxi structure is a real world stochastic multi agent system heavily affected by random external factors. Simulation is a promising means of allowing accurate assessment of the system behavior and consequently its optimal exploitation at almost no risk.

The purpose of this paper is to present a brief overview of a methodology providing an optimal management of such a "Collective Taxi" system by discrete event simulations. With the aim of evaluating the system performances for any possible strategy applied, a discrete event simulator tool has been developed, tested and validated, which enable us to manage the three system approaches (decentralized, centralized and mixed managements). After constructing the necessary decision algorithms concerning the control of the system (management of clients and vehicles), but at this stage only for the decentralized mode, we initiated multiple experiments according to different scenarios.

We were then faced with the challenge of dealing with the enormous information output provided by each experiment. After retrieval of the simulation results, we proceeded to a statistical analysis with the aim of providing some preliminary conclusions characterizing the system performances for the chosen policy. Furthermore, we presented a brief methodology suggesting the choice of optimal values for the parameters involved or reasonable trade-offs to permit satisfactory results.

Future work will aim at the study of the centralized and mixed managements. More precisely, we are interested in developing the control algorithms for these approaches, and subsequently to proceed to a comparison of the system performances for each mode (examining if and when the additional costs of central dispatching are justified, etc.). Moreover, a study employing real data can be envisaged in order to provide persuasive answers for those who still hesitate regarding the effective productivity of "Collective Taxis".

Acknowledgment

The authors are indebted to Dr. Frédéric Meunier of LVMT-ENPC for interesting discussions about routing algorithms, and in particular for pointing out the reference [36].

References

[1] E. Lioris, G. Cohen, and A. de La Fortelle, "Evaluation of collective taxi systems by discrete-event simulation," in *The Second International Conference on Advances in System Simulation (SIMUL 2010)*, Nice, France, August 22-27 2010.
[2] "Share taxi," http://en.wikipedia.org/wiki/Share_taxi.
[3] B. Mayhew, *Central Asia*. Lonely Planet, 2007.
[4] A. Bernstein, "NYC taxi commish setting up shared cabs from U.S. open," May 2011, http://transportationnation.org/2011/05/20/nyc-taxi-commish-setting-up-shared-cabs-from-u-s-open/.
[5] "Bay area taxi cab," http://www.bayareataxi.us/services.php.
[6] "Prime time shuttle," http://www.primetimeshuttle.com/.
[7] "Super Shuttle," http://www.supershuttle.com/.

[8] "Train taxi," http://www.holland.com/uk/gettingthere/around/train/traintaxi.jsp.

[9] D. Dufour, "Shared taxi in Brussels: the missing link in urban transport?" 2008, http://www.urbanicity.org/Site/Articles/Dufour.aspx.

[10] "TaxiShareUK," http://www.taxishareuk.com/.

[11] "Rural transport 2000, transport solutions to sustain the rural economy," http://www.buergerbusse-in-deutschland.de/03_vortraege/virgil_differentiated-forms-of-service-in-germany.htm.

[12] P. Lalos, A. Korres, C. K. Datsikas, G. Tombras, and K. Peppas, "A framework for dynamic car and taxi pools with the use of positioning systems," in *2009 Computation World*, Athens, Greece, November 15-20 2009.

[13] C. Tao, "Dynamic taxi-sharing service using intelligent transportation system technologies," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007)*, Shanghai, China, 21-25 September 2007, pp. 3209–3212.

[14] F. Ciari, M. Balmer, and K. Axhausen, "Large scale use of collective taxis: a multi-agent approach," in *12th International Conference on Travel Behaviour Research*, Jaipur, India, December 2009.

[15] J. Xu and Z. Huang, "Autonomous dial-a-ride transit introductory overview," *Journal of Software*, vol. 4, no. 7, pp. 766–776, September 2009.

[16] P. Fargier and G. Cohen, "Study of a collective taxi system," in *Proceedings of the Fifth International Symposium on the Theory of Traffic Flow and Transportation*, G. Newell, Ed. University of California, Berkeley: American Elsevier, June 16–18 1971, pp. 361–376.

[17] D. W. Lee, E. J. Bass, S. D. Patek, and J. A. Boyd, "A traffic engineering model for air taxi services," *Transportation Research E: The Logistics and Transportation Review*, vol. 44, pp. 1139–1161, 2008.

[18] M. Barth and M. Todd, "Simulation model performance analysis of a multiple station shared vehicle system," *Transportation Research Part C: Emerging Technologies*, vol. 7, pp. 237–259, 1999.

[19] R. Keen and J. Spain, *Computer simulation in biology, a basic introduction*. John Wiley & Sons, 1992, 2nd edition.

[20] E. Gloaguen, C. Dubreuil-Boisclair, P. Simard, B. Giroux, and D. Marcotte, "Simulation of porosity field using wavelet Bayesian inversion of crosswell GPR and log data," in *XIIIth International Conference on Ground Penetrating Radar*, Lecce, Italy, 21-25 June 2010, pp. 598–603.

[21] "US naval academy uses simulation solutions," http://navaltoday.com/2011/03/11/us-naval-academy-uses-simulation-solutions/.

[22] "Simulation," http://en.wikipedia.org/wiki/Simulation.

[23] Y. Wu, "Agent behaviour in peer-to-peer shared ride systems," Master's thesis, Departement of Geomatics, The University of Melbourne, Australia, 2007, http://people.eng.unimelb.edu.au/winter/pubs/wu07agent.pdf.

[24] M. Mes, M. van der Heijden, and A. van Harten, "Comparison of agent-based scheduling to look-ahead heuristics for real-time transportation problems," *European Journal of Operational Research*, vol. 181, no. 1, pp. 59–75, 2007.

[25] H. Klee and R. Allen, *Simulation of dynamic systems with MATLAB and Simulink*. CRC Press Inc., 2011, 2nd edition.

[26] N. Matloff, "A discrete-event simulation course based on the SimPy language," 2010, http://heather.cs.ucdavis.edu/~matloff/simcourse.html.

[27] D. Bollier and A. Eliëns, "SIM: a C++ library for discrete event simulation," 1995, http://www.cs.vu.nl/~eliens/sim/sim_html/sim.html.

[28] E. Kofman, M. Lapadula, and E. Pagliero, "Powerdevs: A devs-based environment for hybrid system modeling and simulation," FCEIA, National University of Rosario, Tech. Rep., 2003, http://www.fceia.unr.edu.ar/~kofman/files/lsd0306.pdf.

[29] "Tortuga (software)," http://en.wikipedia.org/wiki/Tortuga_(software).

[30] A. Varol and M. Günal, "SharpSim tutorial-1," May 2011, http://sharpsim.codeplex.com/.

[31] A. Borshchev, "Simulation modeling with AnyLogic: Agent based, discrete event and system dynamics methods," 2011, http://www.xjtek.com/anylogic/resources/book/.

[32] W. David Kelton, R. Sadowski, and D. Sturrock, *Simulation with Arena*. McGraw-Hill Higher Education, 2003.

[33] K. Concannon, M. Elder, K. Hindle, J. Tremble, and S. Tse, *Simulation Modeling with SIMUL8*. Visual Thinking International, 2007, 4th edition.

[34] "Simcad Pro," http://en.wikipedia.org/wiki/Simcad_Pro.

[35] "Lanner group ltd," http://en.wikipedia.org/wiki/Lanner_Group_Ltd.

[36] J. Tsitsiklis, "Special cases of travelling salesman and repairman problems with time windows," *Networks*, vol. 22, pp. 263–282, 1992.

[37] S. Gaubert, "Howard's multichain policy iteration algorithm for max-plus linear maps." [Online]. Available: http://amadeus.inria.fr/gaubert/HOWARD2.html

[38] J. Cochet-Terrasson and S. Gaubert, "Policy iteration algorithm for shortest path problems," 2000, http://amadeus.inria.fr/gaubert/PAPERS/shortestpath.ps.

[39] C. Bunks, J. Chancelier, F. Delebecque, C. Gomez, M. Goursat, R. Nikoukhah, and S. Steer, *Engineering and Scientific Computing with Scilab*. Boston: Birkhäuser, 1999.

[40] C. Gomez and M. Goursat, "METANET: a system for network problems study," INRIA, Rocquencourt, Le Chesnay, France, Tech. Rep. 124, Novembre 1990, http://hal.inria.fr/docs/00/07/00/43/PDF/RT-0124.pdf.

[41] E. Lioris, "Évaluation et optimisation de systêmes de taxis collectifs en simulation," Ph.D. dissertation, École des Ponts-ParisTech, Marne la Vallée, France, December 2010.

Asymptotically Valid Confidence Intervals for Quantiles and Values-at-Risk When Applying Latin Hypercube Sampling

Marvin K. Nakayama
Computer Science Department
New Jersey Institute of Technology
Newark, New Jersey, 07102, USA
marvin@njit.edu

***Abstract*—Quantiles, which are also known as values-at-risk in finance, are often used as risk measures. Latin hypercube sampling (LHS) is a variance-reduction technique (VRT) that induces correlation among the generated samples in such a way as to increase efficiency under certain conditions; it can be thought of as an extension of stratified sampling in multiple dimensions. This paper develops asymptotically valid confidence intervals for quantiles that are estimated via simulation using LHS.**

***Keywords*-quantile; value-at-risk; Latin hypercube sampling; variance reduction; confidence interval.**

I. INTRODUCTION

Complex stochastic systems arise in many application areas, such as supply-chain management, transportation, networking, and finance. The size and complexity of such systems often preclude the availability of analytical methods for studying the resulting stochastic models, so simulation is frequently used.

Suppose that we are interested in analyzing the behavior of a system over a (possibly random) finite time horizon, and let X be a random variable denoting the system's (random) performance over the time interval of interest. For example, X may represent the time to complete a project, or X may be the loss of a portfolio of financial investments over the next two weeks. Most simulation textbooks focus on estimating the mean μ of X. This typically involves running independent and identically distributed (i.i.d.) replications of the system over the time horizon, and estimating μ via the sample average of the outputted performance from the replications. To provide a measure of the error in the estimate of μ, the analyst will often construct a confidence interval for μ using the simulated output; e.g., see Section 9.4.1 of [2].

In many contexts, however, performance measures other than a mean provide more useful information. One such measure is a *quantile*. For $0 < p < 1$, the p-quantile ξ_p of a random variable X is the smallest constant x such that $P(X \leq x) \geq p$. A well-known example is the median, which the 0.5-quantile. In terms of the cumulative distribution function (CDF) F of X, we can write $\xi_p = F^{-1}(p)$. Quantiles arise in many practical situations, often to measure risk. For example, in bidding on a project, a contractor may want to determine a date such that his firm has a 95% chance of finishing the project by that date, which is the 0.95-quantile. In finance, quantiles, which are known as values-at-risk, are frequently used as measures of risk of portfolios of assets [3]. For example, a portfolio manager may want to know the 0.99-quantile $\xi_{0.99}$ of the loss of his portfolio over the next two weeks, so there is a 1% chance that the loss over the next two weeks will exceed $\xi_{0.99}$.

Estimation of a quantile ξ_p is complicated by the fact that ξ_p cannot be expressed as the mean of a random variable, so one cannot estimate ξ_p via a sample average. However, the fact that $\xi_p = F^{-1}(p)$ suggests an alternative approach: develop an estimator of the CDF F, which may be a sample average, and then invert the estimated CDF.

In addition to a point estimator of ξ_p, we also would like a confidence interval (CI) of ξ_p to provide a measure of the accuracy of the point estimator. One approach to developing a CI is to first prove that the estimator of ξ_p satisfies a central limit theorem (CLT), and then construct a consistent estimator of the variance constant appearing in the CLT to obtain a CI.

Sometimes, the CI for quantile ξ_p is large, especially when $p \approx 0$ or $p \approx 1$, motivating the use of a variance-reduction technique (VRT) to obtain a quantile estimator with smaller error. VRTs that have been applied to quantile estimation include control variates (CV) [4], [5]; induced correlation, including antithetic variates (AV) and Latin hypercube sampling (LHS) [6], [7]; importance sampling (IS) [8]; and combined importance sampling and stratified sampling (IS+SS) [9]. Typically, variance reduction for quantile estimation entails applying a VRT to estimate the CDF, and then inverting the resulting CDF estimator to obtain a quantile estimator.

While most of the papers in the previous paragraph establish CLTs for the corresponding quantile estimators when applying VRTs, none of them provides a way to consistently estimate the CLTs' variance constants. Indeed, [9] states that this is "difficult and beyond the scope of this paper." To address this issue, [10] develops a general framework for analyzing some asymptotic properties (as the sample size gets large) of quantile estimators when applying

VRTs, and [10] shows how this can be exploited to construct consistent estimators of the variance constants in the CLTs. Also, [10] shows the framework encompasses CV, IS+SS and AV. In the current paper, we now do the same for LHS.

The rest of the paper is organized as follows. Section II develops the mathematical framework. We describe LHS in Section III, giving a CI for a quantile estimated via LHS. We present experimental results on a small example in Section IV. Section V provides some concluding remarks, and Section VI contains the proofs of our theorems. The current paper is based on and expands a previous conference paper [1], which includes neither the experimental results nor the proofs.

II. Background

Consider a random variable X having CDF F. For fixed $0 < p < 1$, the goal is to estimate the p-quantile $\xi_p = F^{-1}(p)$ of X, where $F^{-1}(q) = \inf\{x : F(x) \geq q\}$ for any $0 < q < 1$. We assume that X can be expressed as

$$X = g(U_1, U_2, \ldots, U_d) \tag{1}$$

for a known and given function $g : \Re^d \to \Re$, where $U_1, U_2, \ldots, U_d$ are i.i.d. unif$[0, 1)$ random numbers. Thus, generating an output X can be accomplished by transforming d i.i.d. uniforms through g. We now provide examples fitting in this framework.

Example 1: Suppose X is the time to complete a project, and we are interested in computing the 0.95-quantile $\xi_{0.95}$ of X. Assume the time to complete the project is modeled as a stochastic activity network (SAN) [11] having s activities, labeled $1, \ldots, s$. Suppose that there are r paths through the SAN, and let B_j be the set of activities on path j, $j = 1, 2, \ldots, r$. For each activity $i = 1, \ldots, s$, let A_i be its (random) duration. We allow for $A_1, \ldots, A_s$ to be dependent, and let H denote the joint distribution of $A \equiv (A_1, \ldots, A_s)$. Suppose that we can generate a sample of A from H using a fixed number d of i.i.d. unif$[0, 1)$ random variables $U_1, \ldots, U_d$. In the case when $A_1, \ldots, A_s$ are independent with each A_i having marginal distribution H_i, we can generate A_i as $A_i = H_i^{-1}(U_i)$, for $i = 1, \ldots, s$, assuming that H_i^{-1} can be computed efficiently. The length of the jth path in the SAN is $T_j = \sum_{i \in B_j} A_i$, and we can express $X = \max(T_1, \ldots, T_r)$ as the time to complete the project. Thus, the function g in (1) in this case takes the i.i.d. uniforms $U_1, \ldots, U_d$ as arguments, transforms them into $A_1, \ldots, A_s$, computes the length of each path T_j, and returns the maximum path length as X.

Example 2: Consider a financial portfolio consisting of a mix of investments, e.g., stocks, bonds and derivatives. Let $V(t)$ be the value of the portfolio at time t, and suppose the current time is $t = 0$. Let T denote two weeks, and we are interested in the 0.99-quantile of the loss X in the portfolio at the end of this period. Assume we have a stochastic model for $V(T)$, and suppose that simulating $V(T)$ given the current portfolio value $V(0)$ requires generating a fixed number d of i.i.d. unif$[0, 1)$ random variables $U_1, \ldots, U_d$; see Chapter 3 of [12] for algorithms to simulate $V(T)$ under various stochastic models describing the change in values of the investments. Thus, the function g in (1) takes the i.i.d. uniforms $U_1, \ldots, U_d$ as input, transforms them into $V(T)$, and then outputs $X = V(0) - V(T)$ as the portfolio loss. (A negative loss is a gain.) The 0.99-level value-at-risk is then the 0.99-quantile of X.

We now review how quantiles can be estimated when applying *crude Monte Carlo* (CMC) (i.e., no variance reduction). We first generate $n \times d$ i.i.d. unif$[0, 1)$ random numbers $U_{i,j}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, d$, which we arrange in an $n \times d$ grid:

$$\begin{array}{|cccc|} \hline U_{1,1} & U_{1,2} & \cdots & U_{1,d} \\ U_{2,1} & U_{2,2} & \cdots & U_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n,1} & U_{n,2} & \cdots & U_{n,d} \\ \hline \end{array} \; . \tag{2}$$

Then we use the uniforms to generate n outputs $X_1, X_2, \ldots, X_n$ as follows:

$$\begin{array}{|rcl|} \hline X_1 &=& g(U_{1,1}, U_{1,2}, \ldots, U_{1,d}) \\ X_2 &=& g(U_{2,1}, U_{2,2}, \ldots, U_{2,d}) \\ & \vdots & \\ X_n &=& g(U_{n,1}, U_{n,2}, \ldots, U_{n,d}) \\ \hline \end{array} \; .$$

Thus, the ith row of uniforms $U_{i,1}, U_{i,2}, \ldots, U_{i,d}$ from (2) is used to generate the ith output X_i, and the independence of the columns of uniforms in (2) ensures that each X_i has the correct distribution F. Also, because of the independence of the rows of uniforms in (2), we have that $X_1, X_2, \ldots, X_n$ are i.i.d. We then estimate F via the *empirical CDF* $\hat{F}_n$, which is constructed as

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} I(X_i \leq x),$$

where $I(A)$ is the indicator function of the event A, which takes the value 1 (resp., 0) if the event A occurs (resp., does not occur). Then

$$\hat{\xi}_{p,n} = \hat{F}_n^{-1}(p) \tag{3}$$

is the CMC estimator of the p-quantile ξ_p.

Let f denote the derivative of the CDF F, when it exists, and assume F is differentiable at ξ_p with $f(\xi_p) > 0$. The estimator $\hat{\xi}_{p,n}$ then is strongly consistent; i.e., $\hat{\xi}_{p,n} \to \xi_p$ as $n \to \infty$ with probability 1 (e.g., see p. 75 of [13]). Also, $\hat{\xi}_{p,n}$ satisfies the following CLT (Section 2.3.3 of [13]):

$$\frac{\sqrt{n}}{\kappa_p} \left(\hat{\xi}_{p,n} - \xi_p \right) \Rightarrow N(0, 1) \tag{4}$$

as $n \to \infty$, where $\Rightarrow$ denotes convergence in distribution (p. 8 of [13]) and $N(a, b^2)$ is a normal random variable

with mean a and variance b^2. The constant

$$\kappa_p = \sqrt{p(1-p)}\phi_p, \tag{5}$$

where

$$\phi_p = \frac{1}{f(\xi_p)}. \tag{6}$$

Using (4), we can construct an approximate $100(1-\alpha)\%$ CI for ξ_p as

$$\left[\hat{\xi}_{p,n} \pm z_{\alpha/2}\frac{\sqrt{p(1-p)}\,\phi_p}{\sqrt{n}}\right],$$

where $z_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$ and Φ is the CDF of a $N(0,1)$. Unfortunately, the above CI is not implementable in practice since ϕ_p is unknown. Thus, we require a (weakly) consistent estimator of ϕ_p, and for the CMC case, such estimators have been proposed in the statistics literature [14]–[17]. As $\phi_p = \frac{d}{dp}F^{-1}(p)$, these papers develop finite-difference estimators of the derivative (Section 7.1 of [12]):

$$\hat{\phi}_{p,n}(h') = \frac{\hat{F}_n^{-1}(p+h') - \hat{F}_n^{-1}(p-h')}{2h'}, \tag{7}$$

where $h' \equiv h'_n > 0$ is known as the *smoothing parameter*. Consistency holds (i.e., $\hat{\phi}_{p,n}(h'_n) \Rightarrow \phi_p$ as $n \to \infty$) when $h'_n \to 0$ and $nh'_n \to \infty$ as $n \to \infty$. In this case, we obtain the following approximate $100(1-\alpha)\%$ CI for ξ_p:

$$J_n(h'_n) \equiv \left[\hat{\xi}_{p,n} \pm z_{\alpha/2}\frac{\sqrt{p(1-p)}\,\hat{\phi}_{p,n}(h'_n)}{\sqrt{n}}\right], \tag{8}$$

which is asymptotically valid in the sense that

$$P\{\xi_p \in J_n(h'_n)\} \to 1-\alpha$$

as $n \to \infty$. The consistency proofs of $\hat{\phi}_{p,n}(h'_n)$ in [15] and [16] utilize the fact that the i.i.d. outputs X_i, $i = 1, 2, \ldots, n$, can be represented as $X_i = F^{-1}(U_i)$ for $U_i \sim$ unif$[0,1)$ i.i.d. and then exploits properties of uniform order statistics. But this approach does not work when applying VRTs, including LHS, so we require another proof technique when applying LHS.

In the case of CMC, there has been some asymptotic analysis suggesting how one should choose the smoothing parameter $h' = h'_n$ in (7). To asymptotically minimize the mean square error (MSE) of $\hat{\phi}_{p,n}(h'_n)$ as an estimator of ϕ_p, one should choose $h'_n = O(n^{-1/5})$; see [16]. Alternatively, if one wants to minimize the coverage error of the confidence interval in (8), then [17] gives the asymptotically optimal rate for h'_n to be $O(n^{-1/3})$.

III. Latin Hypercube Sampling

LHS, which was introduced by [18] and further analyzed by [19], is a VRT that induces correlation among the simulated outputs in such a way as to increase the statistical efficiency under certain conditions. It can be viewed as an extension of stratified sampling (Section 4.3 of [12]). We next describe an approach in [6] for applying LHS to estimate the quantile ξ_p of the random variable X.

A. Single-Sample LHS

We will first generate a Latin hypercube (LH) sample of size t in dimension d as follows. Let $U_{i,j}$ for $i = 1, \ldots, t$ and $j = 1, \ldots, d$ be $t \times d$ i.i.d. unif$[0,1)$ random numbers. Let $\pi_1, \ldots, \pi_d$ be d independent random permutations of $\{1, \ldots, t\}$; i.e., for each π_j, each of the $t!$ permutations is equally likely. The $\pi_1, \ldots, \pi_d$ are generated independently of the $U_{i,j}$. For each $j = 1, 2, \ldots, d$, we have $\pi_j = (\pi_j(1), \pi_j(2), \ldots, \pi_j(t))$, and $\pi_j(i)$ is the number to which $i \in \{1, 2, \ldots, t\}$ is mapped in the jth permutation. Then define

$$V_{i,j} = \frac{\pi_j(i) - 1 + U_{i,j}}{t};\ i = 1, \ldots, t;\ j = 1, \ldots, d;$$

and arrange them into a $t \times d$ grid:

$$\begin{bmatrix} V_{1,1} & V_{1,2} & \cdots & V_{1,d} \\ V_{2,1} & V_{2,2} & \cdots & V_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ V_{t,1} & V_{t,2} & \cdots & V_{t,d} \end{bmatrix}. \tag{9}$$

For each $i = 1, \ldots, t$, it is easy to show that the ith row $V_i \equiv (V_{i,1}, V_{i,2}, \ldots, V_{i,d})$ from (9) is a vector of d i.i.d. unif$[0,1)$ numbers. But within each column j, the t uniforms $V_{1,j}, V_{2,j}, \ldots, V_{t,j}$ are dependent since they all use the same permutation π_j. Thus, the rows $V_1, V_2, \ldots, V_t$ are dependent, and we call the $t \times d$ uniforms in (9) an *LH sample* of size t in d dimensions.

The LH sample in (9) has an interesting feature, as we now explain. Partition the unit interval $[0,1)$ into t equal-length subintervals $[0, 1/t), [1/t, 2/t), \ldots, [(t-1)/t, 1)$. Then one can show that the t uniforms in any column of (9) have the property that exactly one of them lies in each subinterval. Each of the d columns thus forms a stratified sample of the unit interval in one dimension, so the LH sample can be seen as an extension of stratified sampling in d dimensions.

We use the ith row V_i from (9) to generate an output X'_i:

$$\begin{array}{rcl} X'_1 &=& g(V_{1,1}, V_{1,2}, \ldots, V_{1,d}) \\ X'_2 &=& g(V_{2,1}, V_{2,2}, \ldots, V_{2,d}) \\ & \vdots & \\ X'_t &=& g(V_{t,1}, V_{t,2}, \ldots, V_{t,d}) \end{array}. \tag{10}$$

Each X'_i has distribution F since $V_{i,1}, V_{i,2}, \ldots, V_{i,d}$ from the ith row of (9) are i.i.d. unif$[0,1)$. An estimator of F is then

$$\bar{F}_t(x) = \frac{1}{t}\sum_{i=1}^{t} I(X'_i \le x).$$

We can then invert this to obtain

$$\bar{\xi}_{p,t} = \bar{F}_{p,t}^{-1}(p),$$

which we call the *single-sample LHS (SS-LHS) quantile estimator*. As shown in [6], the estimator $\bar{\xi}_{p,t}$ satisfies the CLT

$$\frac{\sqrt{t}}{\eta_p}(\bar{\xi}_{p,t} - \xi) \Rightarrow N(0,1) \tag{11}$$

as $t \to \infty$ under certain regularity conditions, where

$$\eta_p = \zeta_p \phi_p, \tag{12}$$

ζ_p is given in [6], and ϕ_p is defined in (6). Recalling (4) and (5), which are for the case of CMC, we see that SS-LHS yields an asymptotic variance reduction when $\zeta_p < \sqrt{p(1-p)}$, and [6] provides sufficient conditions to ensure this holds.

Constructing a confidence interval for ξ_p based on the CLT (11) requires consistently estimating η_p. But the dependence of the rows $V_1, V_2, \ldots, V_t$ implies that $X'_1, X'_2, \ldots, X'_t$ are dependent, and this complicates constructing an estimator for η_p. Indeed, [6] does not develop estimators for ζ_p and ϕ_p.

B. Combined Multiple-LHS

To avoid the above complication, rather than taking a single LH sample of size t to obtain a set of t (dependent) outputs as in [6], we instead generate a total of $n = mt$ outputs in groups of t, where each group is constructed from an LH sample of size t and the m different groups are sampled independently. We then use all n samples to compute a CDF estimator, which we invert to obtain a quantile estimator. We incur some loss in statistical efficiency by using m different independent LH samples, each of size t, instead of taking one big LH sample of size n, but [19] notes the degradation is small when t/d is large.

We now provide details of our approach. Define $t \times d \times m$ i.i.d. unif$[0,1)$ random variables $U_{i,j}^{(k)}$, where $i = 1, \ldots, t$; $j = 1, \ldots, d$; and $k = 1, \ldots, m$. Also, let $\pi_j^{(k)}$ for $j = 1, \ldots, d$ and $k = 1, \ldots, m$ be $d \times m$ independent permutations of $\{1, \ldots, t\}$, and the $U_{i,j}^{(k)}$ and the $\pi_j^{(k)}$ are all mutually independent. Each $\pi_j^{(k)} = (\pi_j^{(k)}(1), \pi_j^{(k)}(2), \ldots, \pi_j^{(k)}(t))$, and $\pi_j^{(k)}(i)$ is the value to which i is mapped in permutation $\pi_j^{(k)}$. For each $k = 1, \ldots, m$, let

$$V_{i,j}^{(k)} = \frac{\pi_j^{(k)}(i) - 1 + U_{i,j}^{(k)}}{t};\ i = 1, \ldots, t;\ j = 1, \ldots, d;$$

and we arrange them in a $t \times d$ grid:

$$\begin{bmatrix} V_{1,1}^{(k)} & V_{1,2}^{(k)} & \cdots & V_{1,d}^{(k)} \\ V_{2,1}^{(k)} & V_{2,2}^{(k)} & \cdots & V_{2,d}^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ V_{t,1}^{(k)} & V_{t,2}^{(k)} & \cdots & V_{t,d}^{(k)} \end{bmatrix}, \tag{13}$$

which is an LH sample of size t in d dimensions. We have m such independent grids. Thus, for each grid $k = 1, \ldots, m$, and for each $i = 1, \ldots, t$, we have that $V_i^{(k)} \equiv (V_{i,1}^{(k)}, V_{i,2}^{(k)}, \ldots, V_{i,d}^{(k)})$ is a vector of d i.i.d. unif$[0,1)$ numbers. Also, for each $k = 1, \ldots, m$, the t vectors $V_1^{(k)}, V_2^{(k)}, \ldots, V_t^{(k)}$ are dependent. We use the ith row $V_i^{(k)}$ from (13) to generate an output $X_i^{(k)}$:

$$\begin{array}{|rcl|} \hline X_1^{(k)} &=& g(V_{1,1}^{(k)}, V_{1,2}^{(k)}, \ldots, V_{1,d}^{(k)}) \\ X_2^{(k)} &=& g(V_{2,1}^{(k)}, V_{2,2}^{(k)}, \ldots, V_{2,d}^{(k)}) \\ & \vdots & \\ X_t^{(k)} &=& g(V_{t,1}^{(k)}, V_{t,2}^{(k)}, \ldots, V_{t,d}^{(k)}) \\ \hline \end{array}. \tag{14}$$

Each $X_i^{(k)}$ has distribution F since $V_{i,1}^{(k)}, V_{i,2}^{(k)}, \ldots, V_{i,d}^{(k)}$ from the ith row of (13) are i.i.d. unif$[0,1)$.

Since we independently repeat (14) for $k = 1, 2, \ldots, m$, we get $t \times m$ outputs, which we arrange in a grid:

$$\begin{bmatrix} \boxed{\begin{matrix} X_1^{(1)} \\ X_2^{(1)} \\ \vdots \\ X_t^{(1)} \end{matrix}} & \boxed{\begin{matrix} X_1^{(2)} \\ X_2^{(2)} \\ \vdots \\ X_t^{(2)} \end{matrix}} & \cdots & \boxed{\begin{matrix} X_1^{(m)} \\ X_2^{(m)} \\ \vdots \\ X_t^{(m)} \end{matrix}} \end{bmatrix}. \tag{15}$$

Each boxed column k corresponds to one set of t (dependent) outputs from an LH sample of size t as in (14), so the rows of (15) are dependent. But since we generate the m LH samples independently, the columns of (15) are independent. We subsequently form an estimator of the CDF F as

$$\tilde{F}_{m,t}(x) = \frac{1}{m} \sum_{k=1}^{m} \frac{1}{t} \sum_{i=1}^{t} I(X_i^{(k)} \le x). \tag{16}$$

For any $0 < p < 1$, we then obtain

$$\tilde{\xi}_{p,m,t} = \tilde{F}_{m,t}^{-1}(p), \tag{17}$$

which we call the *combined multiple-LHS (CM-LHS) estimator* of ξ_p.

As we will later see, the CM-LHS quantile estimator $\tilde{\xi}_{p,m,t}$ obeys a CLT, and we will use an approach developed in [10] to estimate the asymptotic variance in the CLT. To do this, first let p_m be any perturbed value of p satisfying $p_m \to p$ as $m \to \infty$, and let $\tilde{\xi}_{p_m,m,t} = \tilde{F}_{m,t}^{-1}(p_m)$. The following theorem, whose proof is in Section VI-A, establishes that $\tilde{\xi}_{p_m,m,t}$ satisfies a so-called Bahadur representation [20].

Theorem 1: If $f(\xi_p) > 0$, then for any $p_m = p + O(m^{-1/2})$,

$$\tilde{\xi}_{p_m,m,t} = \xi'_{p_m} - \frac{\tilde{F}_{m,t}(\xi_p) - p}{f(\xi_p)} + R_m \tag{18}$$

with $\xi'_{p_m} = \xi_p + (p_m - p)/f(\xi_p)$ and

$$\sqrt{m} R_m \Rightarrow 0 \tag{19}$$

as $m \to \infty$. If in addition f is continuous in a neighborhood of ξ_p, then (18)–(19) hold with $\xi'_{p_m} = F^{-1}(p_m)$ for all $p_m \to p$.

A consequence of the Bahadur representation in (18)–(19) is that the CM-LHS quantile estimator $\tilde{\xi}_{p,m,t}$ then satisfies a CLT, which we can see as follows. Take $p_m = p$ in (18), and rearranging terms and multiplying by $\sqrt{m}$ lead to

$$\sqrt{m}(\tilde{\xi}_{p,m,t} - \xi_p) = \sqrt{m}\left(\frac{p - \tilde{F}_{m,t}(\xi_p)}{f(\xi_p)}\right) + \sqrt{m}R_m. \quad (20)$$

Let

$$W^{(k)}(x) = \frac{1}{t}\sum_{i=1}^{t} I(X_i^{(k)} \le x),$$

so $\tilde{F}_{m,t}(x) = \frac{1}{m}\sum_{k=1}^{m} W^{(k)}(x)$. For any fixed x, the $W^{(k)}(x)$, $k = 1, \ldots, m$, are i.i.d., and since each $X_i^{(k)}$ has distribution F, we see that $E[W^{(k)}(x)] = F(x)$. Define

$$\psi_p^2 = \mathrm{Var}[W^{(k)}(\xi_p)],$$

which is finite since $0 \le W^{(k)}(x) \le 1$ for all x. Hence, the ordinary CLT (e.g., p. 28 of [13]) implies that for fixed t,

$$\frac{\sqrt{m}}{\psi_p}\left(F(\xi_p) - \tilde{F}_{m,t}(\xi_p)\right) \Rightarrow N(0,1) \quad (21)$$

as $m \to \infty$. Since $F(\xi_p) = p$, we then get that if $f(\xi_p) > 0$,

$$\frac{\sqrt{m}}{\psi_p / f(\xi_p)}\left(\frac{p - \tilde{F}_{m,t}(\xi_p)}{f(\xi_p)}\right) \Rightarrow N(0,1)$$

as $m \to \infty$ with t fixed. Thus, it follows from (19), (20) and Slutsky's theorem (e.g., p. 19 of [13]) that for fixed t,

$$\frac{\sqrt{m}}{\tau_p}\left(\tilde{\xi}_{p,m,t} - \xi_p\right) \Rightarrow N(0,1) \quad (22)$$

as $m \to \infty$, where

$$\tau_p = \psi_p \phi_p$$

and $\phi_p = 1/f(\xi_p)$, as in (6).

To construct a CI for ξ_p based on the CLT in (22) for the CM-LHS quantile estimator, we now want consistent estimators of ψ_p and ϕ_p. We can estimate ψ_p^2 via

$$\tilde{\psi}_{p,m,t}^2 = \frac{1}{m-1}\sum_{k=1}^{m}\left(W^{(k)}(\tilde{\xi}_{p,m,t}) - \bar{W}_m\right)^2,$$

where

$$\bar{W}_m = \frac{1}{m}\sum_{k=1}^{m} W^{(k)}(\tilde{\xi}_{p,m,t}).$$

Even though $W^{(k)}(x)$, $k = 1, \ldots, m$, are i.i.d. for any fixed x, each $W^{(k)}(\tilde{\xi}_{p,m,t})$ depends on $\tilde{\xi}_{p,m,t}$, which is a function of *all* of the samples by (16) and (17), and this induces dependence among $W^{(k)}(\tilde{\xi}_{p,m,t})$, $k = 1, 2, \ldots, m$. This complicates the analysis of $\tilde{\psi}_{p,m,t}$, but we can apply the techniques in [10] to prove that

$$\tilde{\psi}_{p,m,t} \Rightarrow \psi_p \quad (23)$$

as $m \to \infty$ for fixed $t \ge 1$. An estimator for $\phi_p = \frac{d}{dp}F^{-1}(p)$ is the finite difference

$$\tilde{\phi}_{p,m,t}(h_m) = \frac{\tilde{F}_{m,t}^{-1}(p + h_m) - \tilde{F}_{m,t}^{-1}(p - h_m)}{2h_m} \quad (24)$$

for smoothing parameter $h_m > 0$. In the proof of Theorem 2 below, we prove that

$$\tilde{\phi}_{p,m,t}(h_m) \Rightarrow \phi_p \quad (25)$$

as $m \to \infty$ for fixed $t \ge 1$ under certain conditions (given in Theorem 2) on h_m and the CDF F.

When (23) and (25) hold, Slutsky's theorem guarantees the CLT in (22) remains valid when we replace $\tau_p = \psi_p \phi_p$ with its consistent estimator $\tilde{\psi}_{p,m,t}\tilde{\phi}_{p,m,t}(h_m)$. We then obtain the following approximate $100(1-\alpha)\%$ CI for ξ_p when applying CM-LHS:

$$\tilde{J}_{m,t}(h_m) \equiv \left[\tilde{\xi}_{p,m,t} \pm z_{\alpha/2}\frac{\tilde{\psi}_{p,m,t}\tilde{\phi}_{p,m,t}(h_m)}{\sqrt{m}}\right]. \quad (26)$$

The following theorem, whose proof is in Section VI-B, establishes the asymptotic validity of the above CI.

Theorem 2: If $f(\xi_p) > 0$, then for any fixed $t \ge 1$,

$$\lim_{m\to\infty} P\{\xi_p \in \tilde{J}_{m,t}(h_m)\} = 1 - \alpha \quad (27)$$

for $h_m = cm^{-1/2}$ and any constant $c > 0$. If in addition f is continuous in a neighborhood of ξ_p, then (27) holds for any $h_m > 0$ satisfying $h_m \to 0$ and $1/h_m = O(m^{1/2})$.

The range of valid values for the smoothing parameter h_m in the second case of Theorem 2 is of particular interest since it covers the asymptotically optimal rates for CMC, as discussed at the end of Section II.

We close this section describing how to invert $\tilde{F}_{m,t}$, which is needed to compute the CM-LHS quantile estimator $\tilde{\xi}_{p,m,t}$ in (17) and the finite difference $\tilde{\phi}_{p,m,t}(h_m)$ in (24). First take the $n = mt$ values $X_i^{(k)}$ for $i = 1, \ldots, t$ and $k = 1, \ldots, m$, and sort them in ascending order as $X_{(1)} \le X_{(2)} \le \cdots \le X_{(n)}$. Then for $0 < q < 1$, we can compute $\tilde{F}_{m,t}^{-1}(q) = X_{(\lceil nq \rceil)}$, where $\lceil \cdot \rceil$ is the round-up function.

IV. Experimental Results

We now present some results from running experiments on a small SAN, and below we follow the notation developed in Example 1 from Section II. Previously studied in [4] and [10], the SAN has $s = 5$ activities, where the activity durations are i.i.d. exponential with mean 1. Since the activities are independent, we generate the s activity durations using $d = s$ i.i.d. unif$[0, 1)$ random variables via inversion; e.g., see Section 2.2.1 of [12]. There are $r = 3$ paths in the SAN, with $B_1 = \{1, 2\}$, $B_2 = \{1, 3, 5\}$, and $B_3 = \{4, 5\}$ as the collections of activities on the 3 paths. The goal is to estimate the p-quantile ξ_p of the length X of the longest path for different values of p.

		CM-LHS					
		normal critical point			Student-T critical point		
n	CMC	$t=10$	$t=20$	$t=50$	$t=10$	$t=20$	$t=50$
100	0.899	0.877	0.838	0.618	0.906	0.904	0.739
	(0.359)	(0.229)	(0.212)	(0.171)	(0.255)	(0.275)	(0.658)
400	0.881	0.879	0.867	0.838	0.887	0.883	0.883
	(0.162)	(0.106)	(0.101)	(0.098)	(0.108)	(0.106)	(0.112)
1600	0.885	0.887	0.879	0.879	0.889	0.884	0.889
	(0.081)	(0.053)	(0.051)	(0.050)	(0.053)	(0.052)	(0.052)
6400	0.898	0.895	0.891	0.897	0.895	0.893	0.899
	(0.041)	(0.027)	(0.026)	(0.025)	(0.027)	(0.026)	(0.025)

Table I
COVERAGES (AND AVERAGE HALF WIDTHS) FOR CONFIDENCE INTERVALS USING CMC AND CM-LHS FOR $p = 0.5$

		CM-LHS					
		normal critical point			Student-T critical point		
n	CMC	$t=10$	$t=20$	$t=50$	$t=10$	$t=20$	$t=50$
100	0.868	0.861	0.810	0.549	0.891	0.883	0.616
	(0.706)	(0.578)	(0.512)	(0.362)	(0.644)	(0.663)	(0.391)
400	0.885	0.877	0.869	0.846	0.886	0.885	0.891
	(0.348)	(0.285)	(0.260)	(0.230)	(0.292)	(0.274)	(0.265)
1600	0.899	0.891	0.888	0.878	0.893	0.892	0.889
	(0.173)	(0.142)	(0.130)	(0.117)	(0.143)	(0.131)	(0.121)
6400	0.898	0.902	0.895	0.890	0.903	0.896	0.893
	(0.086)	(0.071)	(0.065)	(0.059)	(0.071)	(0.065)	(0.059)

Table II
COVERAGES (AND AVERAGE HALF WIDTHS) FOR CONFIDENCE INTERVALS USING CMC AND CM-LHS FOR $p = 0.9$

Tables I and II present results for $p = 0.5$ and $p = 0.9$, respectively. In our experiments we constructed nominal $100(1-\alpha)\% = 90\%$ confidence intervals (CIs) for ξ_p based on the CMC and the CM-LHS quantile estimators from (3) and (17), respectively. We ran 10^4 independent replications to estimate coverage levels and average half widths (in parentheses) of the confidence intervals. We varied the total sample size $n = 4^a \times 100$ for $a = 0, 1, 2, 3$. When applying CM-LHS, we varied the LH sample size as $t = 10$, 20 and 50. We set the smoothing parameter for CMC to be $h'_n = 0.5n^{-1/2}$; for CM-LHS, we chose $h_m = 0.5(tm)^{-1/2}$, so $h'_n = h_m$ in all our experiments when $n = mt$. Column 2 in the tables gives the results for the CMC CIs from (8). Columns 3–5 correspond to the CM-LHS CIs in (26), where we recall $z_{\alpha/2}$ is the $1 - \alpha/2$ critical point of a standard normal. For columns 6–8, we replace $z_{\alpha/2}$ in (26) with the $1 - \alpha/2$ critical point $T_{m-1,\alpha/2}$ of a Student-T distribution with $m - 1$ degrees of freedom (df); i.e., if G_{m-1} is the CDF of a Student-T random variable with $m - 1$ df, then $T_{m-1,\alpha/2} = G^{-1}_{m-1}(1 - \alpha/2)$. Since $T_{m-1,\alpha/2} \to z_{\alpha/2}$ as $m \to \infty$, the resulting CI with critical point $T_{m-1,\alpha/2}$ is also asymptotically valid as $m \to \infty$ with fixed t. But $T_{m-1,\alpha/2} > z_{\alpha/2}$ for all m and α, so CIs with Student-T critical points are wider than those with the normal critical point, which can lead to higher coverage.

We first discuss the results for $p = 0.5$ (Table I). CMC gives close to nominal coverage (0.9) for all sample sizes n. The same is true for CM-LHS with $t = 10$ and the normal critical point $z_{\alpha/2}$, but now the average half width decreased by about 35%. For CM-LHS with $t = 20$ and $t = 50$ and the normal critical point, coverage is less than 0.9 for small n, but when n is large, coverage is close to nominal and the average half width is about the same as for $t = 10$. The poor coverage for small n and large t arises because the value of $m = n/t$ is then small; e.g., when $t = 50$ and $n = 100$, then m is only 2. But the validity of our CM-LHS confidence interval in (26) requires $m \to \infty$ (see Theorem 2), so we see poor coverage when t is large and n is small. Coverage is improved for small n when we instead use the Student-T critical point $T_{m-1,\alpha/2}$ to construct the CI, but it is still significantly below the nominal level for $t = 50$. However, this problem goes away when n is large for both the normal and Student-T critical points since then m is also large, so the asymptotic validity takes effect.

The results for $p = 0.9$ (Table II) exhibit similar qualities to those for $p = 0.5$, but there are some important differences. For $p = 0.9$, we see that the amount that CM-LHS decreases the average half width depends on the choice of t. For $t = 10$, the average half width shrunk by about 17% compared to CMC. Average half width is even smaller when using CM-LHS with larger t, with more than 30% reduction for $t = 50$. Thus, while the choice of LH sample size t does not appear to have much affect on the amount of variance reduction when estimating quantiles in the middle of the distribution, it can have a large impact when estimating more extreme quantiles.

Recall that the CM-LHS quantile estimator $\tilde{\xi}_{p,m,t}$ in (17) is computed by inverting $\tilde{F}_{m,t}$ in (16), which depends on *all* $n = mt$ samples generated. An alternative approach is to use *batching*, where we compute m different independent quantile estimates, one from each of the m columns of size t in (15). Since the m columns are i.i.d., we can then compute the average and sample variance of the m i.i.d. quantile estimates to construct a confidence interval for ξ_p. This approach is discussed on p. 242 of [12] for the case of estimating a mean using LHS, and now we provide details on how one could apply batching for estimating a quantile; see also p. 491 of [12]. For each $k = 1, 2, \ldots, m$, let $\dot{\xi}_{p,k,t} = \dot{F}^{-1}_{k,t}(p)$, where

$$\dot{F}_{k,t}(x) = \frac{1}{t} \sum_{i=1}^{t} I(X_i^{(k)} \le x),$$

which is the estimated CDF from using only the kth boxed column of LH samples in (15). Then the *batched LHS quantile estimator* is

$$\bar{\xi}_{p,m,t} = \frac{1}{m} \sum_{k=1}^{m} \dot{\xi}_{p,k,t}, \tag{28}$$

	$p = 0.5$			$p = 0.9$		
n	$t = 10$	$t = 20$	$t = 50$	$t = 10$	$t = 20$	$t = 50$
100	0.587	0.817	0.891	0.437	0.733	0.876
	(0.218)	(0.242)	(0.607)	(0.470)	(0.546)	(1.327)
400	0.093	0.531	0.836	0.042	0.411	0.768
	(0.103)	(0.103)	(0.111)	(0.222)	(0.234)	(0.245)
1600	0.000	0.066	0.652	0.000	0.021	0.487
	(0.051)	(0.050)	(0.051)	(0.110)	(0.114)	(0.113)
6400	0.000	0.000	0.178	0.000	0.000	0.046
	(0.025)	(0.025)	(0.025)	(0.055)	(0.057)	(0.056)

Table III
COVERAGES (AND AVERAGE HALF WIDTHS) FOR LHS CONFIDENCE INTERVALS USING BATCHING WITH $m = n/t$ BATCHES AND LH SAMPLE SIZE t

n	$p = 0.5$	$p = 0.9$
100	0.587	0.437
	(0.218)	(0.470)
400	0.807	0.720
	(0.109)	(0.241)
1600	0.879	0.850
	(0.055)	(0.118)
6400	0.889	0.888
	(0.027)	(0.060)

Table IV
COVERAGES (AND AVERAGE HALF WIDTHS) FOR LHS CONFIDENCE INTERVALS USING BATCHING WITH $m = 10$ BATCHES AND LH SAMPLE SIZE $t = n/m$

and an approximate $100(1-\alpha)\%$ confidence interval for ξ_p is

$$\left[\bar{\xi}_{p,m,t} \pm T_{m-1,\alpha/2}\frac{S_m}{\sqrt{m}}\right], \tag{29}$$

where

$$S_m^2 = \frac{1}{m-1}\sum_{k=1}^{m}(\dot{\xi}_{p,k,t} - \bar{\xi}_{p,m,t})^2.$$

As in our experiments with the CM-LHS quantile estimator in Tables I and II, we also ran experiments with the batched CI in (29) for $p = 0.5$ and 0.9 to study its behavior as the total sample size n increases. Since $n = mt$, increasing n requires correspondingly increasing the number m of batches and/or the LH sample size t.

Table III presents results using batching in which we keep t fixed at 10, 20 or 50 so $m = n/t$ increases as n grows. For $n = 100$, coverage is close to nominal for $t = 50$, but coverage is significantly low for $t = 10$ and $t = 20$. As n increases, coverage actually *worsens* for each t. This occurs because quantile estimators are biased, with bias decreasing as the sample size increases; see [6]. In each column of Table III, t is fixed and does not increase, so $\bar{\xi}_{p,m,t}$ is the average of m biased quantile estimators $\dot{\xi}_{p,k,t}$, $k = 1, 2, \ldots, m$, each computed from a single LH sample of size t, with more bias for small t. If $E[|\dot{\xi}_{p,1,t}|] < \infty$, then

$$\bar{\xi}_{p,m,t} = \frac{1}{m}\sum_{k=1}^{m}\dot{\xi}_{p,k,t} \Rightarrow E[\dot{\xi}_{p,1,t}]$$

as $m \to \infty$ by the law of large numbers since $\dot{\xi}_{p,k,t}$, $k = 1, 2, \ldots, m$, are i.i.d. But $E[\dot{\xi}_{p,1,t}] \neq \xi_p$ because of the bias for fixed t. Thus, as n and m increase, the CI in (29) is shrinking at rate $m^{-1/2}$, but it is centered at an estimate whose bias is not decreasing since t is fixed. This results in the poor coverage. For a batching approach to work, we instead need the LH sample size t to increase as the total sample size n increases to ensure the bias of $\bar{\xi}_{p,m,t}$ decreases.

Table IV presents results with batching in which the number of batches is fixed at $m = 10$, so the LH sample size $t = n/m$ grows as the total sample size n increases. In contrast to the case when t was fixed as n increases (Table III), we now see in Table IV that the coverage levels of the CIs in (29) converge to the nominal level 0.9 as n increases. However, compared to the results in Tables I and II for the CM-LHS quantile estimator, we see that batching with t increasing in n gives lower coverage when n is small. This occurs because of the bias of quantile estimators, as we discussed in the previous paragraph. The batched LHS estimator in (28) averages m i.i.d. quantile estimators, each based on an LH sample of size t, so the bias of the batched quantile estimator is determined by t. On the other hand, CM-LHS computes a single quantile estimator (17) based on inverting the CDF estimator (16) from all of the $n = mt$ samples. Because the bias of quantile estimators decreases as the sample size grows, the batched LHS quantile estimator has larger bias than the CM-LHS quantile estimator. This leads to the coverage of batched CI in (29) converging more slowly to the nominal level as the total sample size n grows than the CI in (26) for CM-LHS. This property is more pronounced for $p = 0.9$ than for $p = 0.5$, so batching seems to do worse for extreme quantiles than for those in the middle of the distribution.

V. CONCLUSION

We presented an asymptotically valid CI for a quantile estimated using LHS. We developed a combined multiple-LHS approach in which one generates a total of n samples in m independent groups, where each group is generated from an LH sample of size t. Using a general framework developed in [10] for quantile estimators from applying VRTs, we proved that the resulting CM-LHS quantile estimator satisfies a Bahadur representation, which provides an asymptotic approximation for the estimator. The Bahadur representation implies a CLT for the CM-LHS quantile estimator and also allows us to construct a consistent estimator for the asymptotic variance in the CLT.

We ran experiments on a small SAN, and our results demonstrate the asymptotic validity of our CM-LHS CIs. Also, the results show that CM-LHS can decrease the aver-

age half width of confidence intervals relative to CMC, but the amount of decrease can depend of the LH sample size t when estimating an extreme quantile. We also experimented with an alternative approach based on batching with m independent batches, each consisting of t samples constructed from an LH sample of size t. To lead to asymptotically valid CIs, batching requires t to increase as the total sample size $n = mt$ grows. Compared to CM-LHS, batching needs larger samples sizes n for the CIs to have close to nominal coverage. Further work is needed to study the empirical performance of the proposed method when simulating other larger stochastic models.

VI. Appendix

A. Proof of Theorem 1

Chu and Nakayama [10] develop a general framework giving sufficient conditions for quantile estimators obtained when applying VRTs to satisfy a Bahadur representation, and we now establish (18)–(19) by showing that our CM-LHS approach fits into the framework. Specifically, for the first result in Theorem 1 (i.e., for $p_m = p + O(m^{-1/2})$), we need to show that $\tilde{F}_{m,t}$ in (16) satisfies the following assumptions from [10]:

Assumption A1: $P(M_{m,t}) \to 1$ as $m \to \infty$, where $M_{m,t}$ is the event that $\tilde{F}_{m,t}(x)$ is monotonically increasing in x for fixed m and t.

Assumption A2: $E[\tilde{F}_{m,t}(x)] = F(x)$ for all x, and for every $a_m = O(m^{-1/2})$,

$$E\left[\tilde{F}_{m,t}(\xi_p + a_m) - \tilde{F}_{m,t}(\xi_p)\right]^2 = [F(\xi_p + a_m) - F(\xi_p)]^2 + s_m(a_m)/m$$

with $s_m(a_m) \to 0$ as $m \to \infty$.

Assumption A3: $\sqrt{m}\left[\tilde{F}_{m,t}(\xi_p) - F(\xi_p)\right] \Rightarrow N(0, \psi_p^2)$ as $n \to \infty$ for some $0 < \psi_p < \infty$.

As shown in [10], if $f(\xi_p) > 0$, then Assumptions A1–A3 imply that (18) and (19) hold for any $p_m = p + O(m^{-1/2})$. Also, [10] proves that if we additionally strengthen A2 to be true for all $a_m \to 0$ and f is continuous in a neighborhood of ξ_p, then (18) and (19) hold for any $p_m \to p$, which is what we need to show for the second part of Theorem 1. Thus, we now prove that $\tilde{F}_{m,t}$ in (16) satisfies A1–A3, with A2 holding for all $a_m \to 0$.

Examining (16), we see that Assumption A1 holds since $\tilde{F}_{m,t}(x)$ is always monotonically increasing in x because each $I(X_i^{(k)} \le x)$ has this property. Also, we previously showed Assumption A3 holds in (21), so it remains to prove Assumption A2 holds, which we show is true for any $a_m \to 0$.

Since each $X_i^{(k)}$ has distribution F, we have that

$$E[\tilde{F}_{m,t}(x)] = \frac{1}{m}\sum_{k=1}^{m}\frac{1}{t}\sum_{i=1}^{t}E[I(X_i^{(k)} \le x)] = F(x)$$

for all x, so the first part of A2 holds. To establish the second part of A2, let $\rho_m = \min(\xi_p, \xi_p + a_m)$ and $\rho'_m = \max(\xi_p, \xi_p + a_m)$ for any $a_m \to 0$. Then

$$b_m \equiv E\left[\tilde{F}_{m,t}(\xi_p + a_m) - \tilde{F}_{m,t}(\xi_p)\right]^2 = E\left(\frac{1}{m}\sum_{k=1}^{m}\frac{1}{t}\sum_{i=1}^{t}C_i^{(k)}\right)^2, \quad (30)$$

where $C_i^{(k)} = I(\rho_m < X_i^{(k)} \le \rho'_m)$. Note that $C_i^{(k)}$ and $C_{i'}^{(k')}$ are independent for $k \ne k'$ and any i and i' since $C_i^{(k)}$ and $C_{i'}^{(k')}$ correspond to outputs from different LH samples. Thus, expanding the square in (30) gives

$$b_m = \frac{1}{(mt)^2}\sum_{k=1}^{m}\sum_{i=1}^{t}E[(C_i^{(k)})^2] + \frac{1}{(mt)^2}\sum_{k=1}^{m}\sum_{\substack{k'=1\\k'\ne k}}^{m}\sum_{i=1}^{t}\sum_{i'=1}^{t}E[C_i^{(k)}]E[C_{i'}^{(k')}] + c_m,$$

where

$$c_m = \frac{1}{(mt)^2}\sum_{k=1}^{m}\sum_{i=1}^{t}\sum_{\substack{i'=1\\i'\ne i}}^{t}E[C_i^{(k)}C_{i'}^{(k)}].$$

For all i and k, we have $E[C_i^{(k)}] = F(\rho'_m) - F(\rho_m) \equiv d_m$ and $(C_i^{(k)})^2 = C_i^{(k)}$. Thus,

$$b_m = \frac{1}{mt}d_m + \frac{m-1}{m}d_m^2 + c_m = d_m^2 + e_m + c_m,$$

where

$$e_m = \frac{1}{mt}d_m - \frac{1}{m}d_m^2.$$

Since $[F(\xi_p + a_m) - F(\xi_p)]^2 = d_m^2$, A2 holds if we show that $me_m \to 0$ and $mc_m \to 0$ as $m \to \infty$.

Now $me_m \to 0$ holds since

$$d_m \to 0 \quad (31)$$

because $a_m \to 0$ and F is continuous at ξ_p. To show $mc_m \to 0$, note that

$$E[C_i^{(k)}C_{i'}^{(k)}] = P(\rho_m < X_i^{(k)} \le \rho'_m,\ \rho_m < X_{i'}^{(k)} \le \rho'_m) \le P(\rho_m < X_i^{(k)} \le \rho'_m) = d_m,$$

so it follows that

$$|mc_m| \le \frac{m}{(mt)^2}\sum_{k=1}^{m}\sum_{j=1}^{t}\sum_{\substack{j'=1\\j'\ne j}}^{t}d_m = \frac{t-1}{t}d_m \to 0$$

as $m \to \infty$ by (31). Thus, the proof is complete.

B. Proof of Theorem 2

All that remains is to prove (25) for the two different cases given in the theorem. For the first case, it is shown in [10] that if $f(\xi_p) > 0$, then it follows from the first Bahadur representation (i.e., with $\xi'_{p_m} = \xi_p + (p_m - p)/f(\xi_p)$) in Theorem 1 that (25) holds when $h_m = cm^{-1/2}$ for any constant $c > 0$. Moreover, [10] shows that when f is continuous in a neighborhood of ξ_p, the second Bahadur representation in Theorem 1 in which $\xi'_{p_m} = F^{-1}(p_m)$ implies (25) holds for any h_m satisfying $h_m \to 0$ and $1/h_m = O(m^{1/2})$, which is what we need to show for the second case of Theorem 2.

ACKNOWLEDGMENT

This work has been supported in part by the National Science Foundation under Grant No. CMMI-0926949. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] M. K. Nakayama, "Confidence Intervals for Quantiles When Applying Latin Hypercube Sampling," in *Proceedings of the Second International Conference on Advances in System Simulation (SIMUL 2010)*, pp. 78–81, 2010.

[2] A. M. Law, *Simulation Modeling and Analysis*, 4th ed. New York: McGraw-Hill, 2006.

[3] D. Duffie and J. Pan, "An overview of value at risk," *Journal of Derivatives*, vol. 4, pp. 7–49, 1997.

[4] J. C. Hsu and B. L. Nelson, "Control variates for quantile estimation," *Management Science*, vol. 36, pp. 835–851, 1990.

[5] T. C. Hesterberg and B. L. Nelson, "Control variates for probability and quantile estimation," *Management Science*, vol. 44, pp. 1295–1312, 1998.

[6] A. N. Avramidis and J. R. Wilson, "Correlation-induction techniques for estimating quantiles in simulation," *Operations Research*, vol. 46, pp. 574–591, 1998.

[7] X. Jin, M. C. Fu, and X. Xiong, "Probabilistic error bounds for simulation quantile estimation," *Management Science*, vol. 49, pp. 230–246, 2003.

[8] P. W. Glynn, "Importance sampling for Monte Carlo estimation of quantiles," in *Mathematical Methods in Stochastic Simulation and Experimental Design: Proceedings of the 2nd St. Petersburg Workshop on Simulation*. Publishing House of St. Petersburg University, St. Petersburg, Russia, 1996, pp. 180–185.

[9] P. Glasserman, P. Heidelberger, and P. Shahabuddin, "Variance reduction techniques for estimating value-at-risk," *Management Science*, vol. 46, pp. 1349–1364, 2000.

[10] F. Chu and M. K. Nakayama, "Confidence intervals for quantiles when applying variance-reduction techniques," *submitted*, 2010.

[11] V. G. Adlakha and V. G. Kulkarni, "A classified bibliography of research on stochastic PERT networks," *INFOR*, vol. 27, pp. 272–296, 1989.

[12] P. Glasserman, *Monte Carlo Methods in Financial Engineering*. New York: Springer, 2004.

[13] R. J. Serfling, *Approximation Theorems of Mathematical Statistics*. New York: John Wiley & Sons, 1980.

[14] M. M. Siddiqui, "Distribution of quantiles in samples from a bivariate population," *Journal of Research of the National Bureau of Standards B*, vol. 64, pp. 145–150, 1960.

[15] D. A. Bloch and J. L. Gastwirth, "On a simple estimate of the reciprocal of the density function," *Annals of Mathematical Statistics*, vol. 39, pp. 1083–1085, 1968.

[16] E. Bofinger, "Estimation of a density function using order statistics," *Australian Journal of Statistics*, vol. 17, pp. 1–7, 1975.

[17] P. Hall and S. J. Sheather, "On the distribution of a Studentized quantile," *Journal of the Royal Statistical Society B*, vol. 50, pp. 381–391, 1988.

[18] M. D. McKay, W. J. Conover, and R. J. Beckman, "A comparison of three methods for selecting input variables in the analysis of output from a computer code," *Technometrics*, vol. 21, pp. 239–245, 1979.

[19] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, pp. 143–151, 1987, correction 32:367.

[20] R. R. Bahadur, "A note on quantiles in large samples," *Annals of Mathematical Statistics*, vol. 37, pp. 577–580, 1966.

A Testing Framework for Assessing Grid and Cloud Infrastructure Interoperability

Thomas Rings, Jens Grabowski
Institute of Computer Science, University of Göttingen
Göttingen, Germany
Email: {rings,grabowski}@cs.uni-goettingen.de

Stephan Schulz
Conformiq Software OY
Espoo, Finland
Email: stephan.schulz@conformiq.com

***Abstract*—The composition of grid and cloud computing infrastructures using equipment from different vendors to allow service enrichment and increase productivity is an important need in industry and for governmental institutions. Interoperability between equipment can be achieved using the gateway approach or the standardized interface approach. These approaches, as well as equipment need to be engineered and developed with the goal to allow problem-free interoperations between involved equipment. A step towards such interoperation is the assessment of interoperability. Focusing on technical interoperability, we present a testing framework for the assessment of interoperability of grid and cloud computing infrastructures. This also includes the assessment of application deployment onto several infrastructures provided by different vendors, which is a key driver for market success. This testing framework is part of an initiative for standardizing the use of grid and cloud technology in the context of telecommunication at the *European Telecommunications Standards Institute* (ETSI). Following the test development process developed and used at ETSI, we developed a test architecture, test configurations, compliance levels, test purposes, interoperability test descriptions, test applications, and a test selection method that together build the testing framework. Its application is exemplified by the assessment of resource reservation and application deployment onto grid and cloud infrastructures based on standardized Grid Component Model descriptors. The presented testing framework has been applied successfully in an interoperability event. In this article, we present a testing framework for the assessment of the interoperability of grid and cloud infrastructure.**

***Keywords*-standardization; interoperability; GCM; grid; cloud; testing**

I. Introduction

Accessing globally distributed data and computing power independent from locations becomes more and more an obligatory requirement from customers in order to store data and utilize computing power. Systems for efficient usage of idling resources, which are physically located all over the world are needed. Grid computing systems, but also recently cloud computing systems, offer methodologies for achieving such a goal. Both offer services for obtaining, providing and selling computing power on demand. This is especially interesting in application domains where computing power is needed spontaneously and in unpredictable time intervals. Many grid and cloud providers have recently appeared on the market offering their own custom-made solutions to address this need. However, from a customer point of view, it is required to access several systems offered by different providers to use more resources, for example, for replication on different systems, but also to save money choosing the best solution. This allows service enrichment by integrating services only available in another infrastructure and to increase productivity by consuming such extended services. A way to achieve this goal is to make these systems interoperable.

Interoperability can be leveraged to open new markets, to foster innovation, and to enable mass markets. Interoperability allows the creation of new and innovative systems through composition of interoperable systems. Furthermore, it increases system availability and reliability. Interoperability provides a great mean to success. However, equipments from which the systems are composed can be developed by different vendors. These equipments need to be engineered to be interoperable. An interim approach is the gateway solution that allows communication between equipments. A gateway converts messages received by one equipment into a representation understandable by another equipment to allow their interoperation. The long-term approach to achieve interoperability is the implementation of agreed specifications, i.e., standards, which capture requirements and functionality. In addition, they define architectures as well as interfaces and specify protocols to be used for communication via these interfaces. Ideal specifications are independent from implementations and leave space for innovation. Even if specifications are assumed as unambiguous, which is rarely the case, testing is needed to validate that implementations follow the specifications. A further step is to test if implementations are able to interoperate.

Proper testing - similar as specification - requires a well defined process and guidelines to be effective in its application. The purpose of testing frameworks is to define a structured approach to test specifications in a given domain, i.e., what to test as well as how to test certain aspects of a *System Under Test* (SUT). They help to increase the quality of test specifications.

At the *European Telecommunications Standards Institute* (ETSI), an initiative for standardizing the use of grid and cloud computing technology in the context of telecommunication, the *Technical Committee CLOUD* (TC CLOUD)

(previously *Technical Committee GRID* (TC GRID)) [1], has been formed. Under its umbrella, standards and interoperability in grid, cloud and telecommunication systems are analyzed, developed and forwarded. These include the following: analysis of interoperability gaps between grid and cloud [2]; surveys on grid and cloud standards [3]; comparisons between grid, cloud and telecommunication systems [4]; the grid standard *Grid Component Model* (GCM) [5]–[7], analysis of architectural options for combining grid and the *Next Generation Network* (NGN) [8]. As part of this initiative towards standardization, we present an interoperability testing framework for grid and cloud computing systems following the systematic test development process developed and used at ETSI [4]. The testing framework unifies testing in diverse domains such as grids and *Infrastructure as a Service* (IaaS) clouds systems, which are dominated by proprietary interfaces for similar functionalities. The framework should be applied in the focus of interoperability events.

This paper extends our previous work [9] with a discussion on differences and similarities of grid and cloud computing systems. Furthermore, concepts on interoperability are presented and a extended description about test configurations and compliance levels, which belong to the presented test framework is given. As a result of applying the testing framework in a cloud and grid interoperability test event, we added consideration about needs of standardizing cloud computing systems.

This article is structured as follows: In Section II, we present the three forms of cloud computing and discuss briefly the differences between grid and cloud computing systems. Afterwards, in Section III, we discuss types of interoperability and approaches on achieving interoperability in software systems. In Section IV, we consider different types of testing in the context of standardization including conformance and interoperability testing. In Section V, we introduce the ETSI GCM standard, which provides the main context for our testing framework that is illustrated afterwards. This testing framework, our main contribution, is applicable for resource reservation and application deployment in grid and cloud infrastructures. In Section VI, the application of the framework is exemplified by the grid middleware Globus Toolkit and the cloud computing system of Amazon. An event around the presented testing framework organized by ETSI is described in Section VII. Resulting from the event, we discuss standardization needs towards interoperability of cloud computing infrastructures. Afterwards, in Section VIII, we provide an overview and a comparison with related work in the domains of grid and cloud computing. Finally, we conclude with a summary and outlook in Section IX.

II. Cloud Versus Grid Computing

Grid computing has a complementary but independent relationship to cloud computing. Both are highly distributed systems and fulfill needs for large computational power and huge storage resources.

However, cloud systems utilize virtualization to offer a uniform interface to dynamically scalable underlying resources. Such a virtualization layer hides heterogeneity, geographical distribution, and faults of resources. By the nature of virtualization, a cloud system provides an isolated and custom-made environment. Clouds are classified in a layered model containing the following layers from bottom to top as depicted in Figure 1: *Infrastructure as a Service* (IaaS), *Platform as a Service* (PaaS), and *Software as a Service* (SaaS).

In the IaaS layer, the user is responsible for *Virtual Machine* (VM) management. Physical hardware that includes disks, processors, and networks are virtualized and configured according to the needs of customers. It is not needed to purchase or manage physical data center equipment. The benefit is its scalability because resources can be added or removed on demand while continuing business operations in a rapid and highly dynamic way. It targets especially system administrators.

The PaaS layer provides a higher abstraction than the IaaS layer and is usually deployed on the virtualized infrastructure of IaaS. PaaS provides a development platform, e.g., an *Application Programming Interface* (API) to request VMs, which are handled transparently by the PaaS. It can also include databases, message queues, or object data stores. PaaS is especially used by software developers and system architects.

SaaS is the instance on top of the cloud model. It provides the application in the cloud, which is only customizable within limits. It focuses on end-user requirements and allows the end-user to access applications intuitively, e.g., via a web browser. The application or software is used on demand over a high-speed network and runs on VMs, which are deployed on the cloud providers' physical hardware. Therefore, current cloud solutions offer dedicated access, i.e., the cloud customer is bound to a company or institution or requires duplicated effort to repeat the deployment process for additional cloud environments.

In contrast, a grid computing environment aggregates heterogeneous resources offered by different providers. It aims to provide a standard set of services and software that enable the collaborative sharing of federated and geographically distributed computing and storage resources. It provides a security framework for identifying inter-organizational parties (both human and electronic), managing data access and movement, and utilization of remote computing resources.

Grid computing can benefit from the development of cloud computing by harnessing new commercially available computing and storage resources, and by deploying cloud technology on grid-enabled resources to improve the management and reliability of those resources via the virtualization layer.

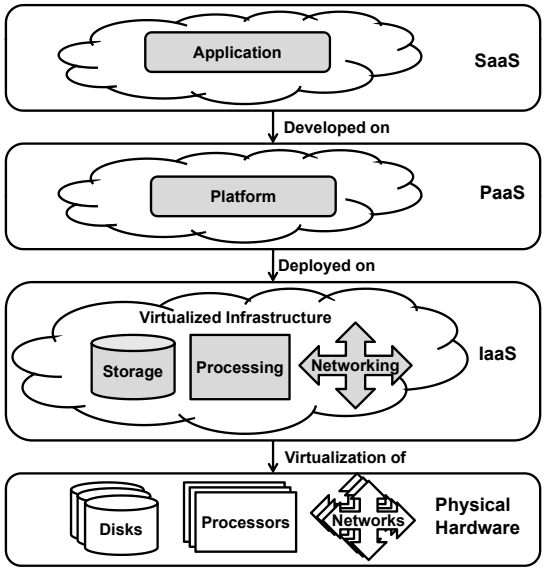

Figure 1. Three Forms of Cloud Computing

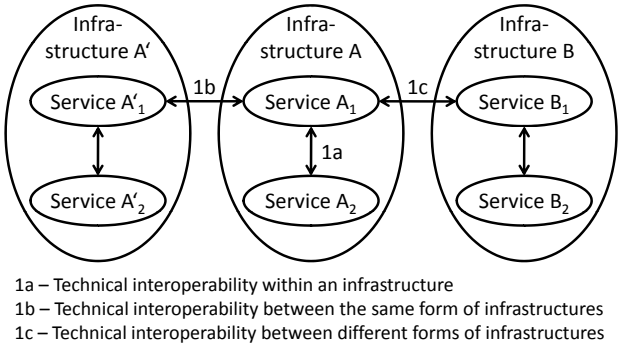

Figure 2. Types of Technical Interoperability

The more advanced state of interface standardization within grid technology allows some degree of choice between various software and hardware systems. Cloud computing still lacks any substantive standards or possibilities for interoperation [2], [4].

The GCM standards, on which the presented testing framework is based on, was originally developed for grid environments. However, it can be extended with clouds. Therefore, the testing framework presented in this article is applicable for grid computing and IaaS cloud computing infrastructures. The GCM standards are described in detail in Section V-A.

III. Achieving Interoperability

Interoperability is crucial to ensure delivery of services across systems from different vendors. To achieve technical interoperability [10], different types according the system interoperation can be identified. This section identifies these types and describes approaches for achieving technical interoperability.

A. Types of Technical Interoperability

Depending on the view on distributed systems, such as grid or cloud computing environments, technical or functional interoperability can be interpreted differently. In general, three different types of technical interoperability including interoperability within an infrastructure, between the same form of infrastructures, and between different forms of infrastructures can be distinguished as depicted in Figure 2 [3].

Interoperability within an infrastructure means that the services provided by an infrastructure or entities using and implementing them are able to communicate by well defined interfaces (Figure 2–1a). This means that the services within a specific infrastructure are able to interoperate through common, standardized (or otherwise agreed) interfaces inside the infrastructure. A practical example is the requirement to utilize two different components such as a billing and a monitoring service implemented by different vendors that need to communicate within one infrastructure.

Interoperability between different infrastructures is usually located at user domain level, i.e., interoperability between end users (Figure 2–1b). An infrastructure A and an infrastructure A' need to be able to communicate and exchange data through one or more standardized interfaces. More specifically, the services provided by infrastructure A understand the services provided by infrastructure A'. For example, a service is able to use an execution service of another infrastructure to reduce computing time. However, this also involves interoperability of other services such as authentication and authorization.

Another type of technical interoperability is interoperability of an infrastructure A with an infrastructure of another form B (Figure 2–1c). Despite other considerations to apply this type, it needs to be determined if the services that need to interoperate for certain functionalities are provided by both infrastructures. The infrastructure should be able to interact in order to exchange information and data, or provide access to resources. For example, a grid system can be extended with storage offered by a cloud computing environment.

Within this paper, we consider technical but also syntactical interoperability between the same form of infrastructures (e.g., grid–grid, cloud–cloud), and between different forms of infrastructures (e.g., cloud–grid).

B. Approaches on Achieving Interoperability

Several approaches to achieve interoperability between computing and storage infrastructures exist. In general, these approaches are classified in gateways and standardized interfaces [3].

A gateway contains several translators and adapters as depicted in Figure 3. The translator transforms data from a

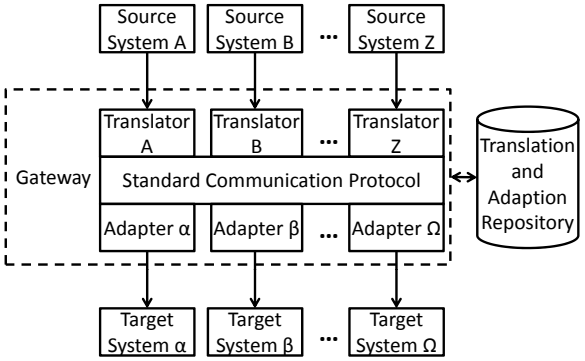

Figure 3. Gateway Approach for Achieving Interoperability

source system into a common and agreed representation, e.g., an *eXtensible Markup Language* (XML) scheme to allow systems using different protocols to be connected to the gateway. The adapter takes this translation and converts it to a specific protocol that is used by the target system. The adapter communicates the translated information to the target system. If the target system replies, it takes the role of a source system. Note that in a one-to-one scenario, it is possible to translate and adapt directly into the required protocol of the target or source system instead of into an agreed intermediate representation. The data of the involved protocols and the translation schemes can be stored in a translation and adaptation repository that can be accessed for translating purposes. Figure 3 shows a specific many-to-many scenario where the gateway resides independent from the involved systems, e.g., in the network. In the one-to-one scenario, the translator and the adapter can reside at the respective system side. Therefore, they do not consolidate a gateway.

Gateways should be considered as interim solutions, as they do not scale well. If the number of systems increases, the gateway performance decreases. It is an expensive approach, because for each protocol, a translator and an adapter need to be developed and integrated. Therefore, gateway solutions are not viable in ad-hoc scenarios or emergency cases.

The long-term approach to address interoperability is the use of open and standardized interfaces. The interfaces that need standardization can evolve from the gateway deployment since the mapping to different infrastructures has already been identified. However, the drawback of this approach is that an agreement on a common set of standard interfaces that also meet production system requirements is very time consuming. However, standardization can enable interoperability in a multi-vendor, multi-network, and multi-service environment without scalability problems as in the gateway approach. Standards need to be engineered for interoperability as described in the next section.

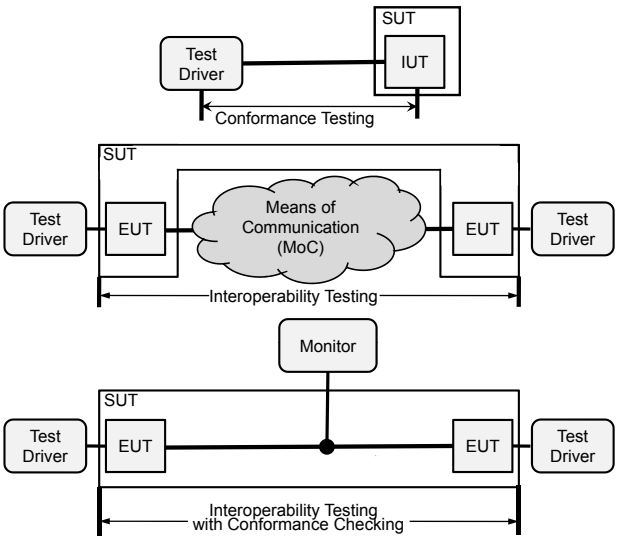

Figure 4. Three approaches to testing in standardization

IV. Conformance vs. Interoperability Testing

Interoperability testing demonstrates that implementations provide end-to-end functionality as described or implied by a specification. In this section, the differences and commonalities among conformance and interoperability testing are described.

Conformance testing is generally used to check that an implementation follows the requirements stated in a specification whereas interoperability testing checks that equipments from different vendors provide an end-to-end service or functionality. In the following, we consider such a specification to be standardized and published by an organization like *Open Grid Forum* (OGF), *World Wide Web Consortium* (W3C) or ETSI.

At ETSI, conformance testing, interoperability testing, or interoperability testing with conformance checking [11] are formally used to test implementations of standards. The three approaches are illustrated in Figure 4. In the following, each approach is considered with its benefits and limitations.

In conformance testing, one *Implementation Under Test* (IUT) is tested with functional black-box tests. Hereby, it is checked if the IUT is conform to a standard. The IUT is embedded by the SUT, which is a testing environment that also includes parts that are required by the IUT to provide its service or functionality to the user. Usually, the development and implementation of sometimes sophisticated testing tools based, e.g., on the *Testing and Test Control Notation* (TTCN-3) [12] is required by conformance testing. Such tools support the simulation of the environment, which is needed for a proper execution of the IUT. However, even if the IUT passes the conformance tests, it does not automatically prove that the IUT is interoperable with other systems implementing the same standard. Standards need to

be engineered for interoperability, because they may contain implementation options and leave space for interpreting requirement specifications, which can lead to interoperability problems.

End-to-end functionality specified or implied by a standard between two or more *Equipment Under Test*s (EUTs) is checked with interoperability testing. Each EUT corresponds to a complete system that can consist of several soft- and hardware components. EUTs interoperate via an abstract *Means of Communication* (MoC). It is generally assumed that the communication services used between EUTs are compliant to underlying standards. Interoperability testing is usually driven manually because of the proprietary nature of end user interfaces and does per se not require any testing tool support. However, from our experiences most interoperability problems in practice are often caused by incorrect or non compliant use of communication interfaces.

To address this problem, ETSI endorses a form of interoperability testing that includes conformance checking, i.e., a hybrid of the two testing approaches. This approach extends end-to-end interoperability testing with the monitoring of the communication among the EUTs. Monitors are used to check the conformance of the EUTs with the relevant protocol specifications within the SUT during the interoperability test. The ETSI experience with applying this hybrid approach during interoperability events [13] is that this approach provides valuable feedback to standardization. Even if end-to-end interoperability has been observed, EUTs did not communicate in a number of cases according to underlying standards. Although this approach is not a replacement for conformance testing, it offers an economic alternative to gain insights about the conformance of equipment participating in an interoperability test to a standard.

V. Testing framework

In this section, we present a testing framework for resource reservation and application deployment onto grid and cloud infrastructures, which is based on the generic ETSI interoperability testing methodology. The framework is based on the ETSI GCM standards [5], [6], which provide a starting point for the extraction of testable requirements for an application deployment. However, the consolidating nature of the GCM standards allows the application of the presented framework outside of the specific context of GCM on other grid and cloud infrastructures. The presented testing framework includes interoperability test configurations and test descriptions, which provide a basis for test specifications that can be applied in the context of an interoperability event. The presented testing framework can be used to assess interoperability within and between grid and cloud infrastructures.

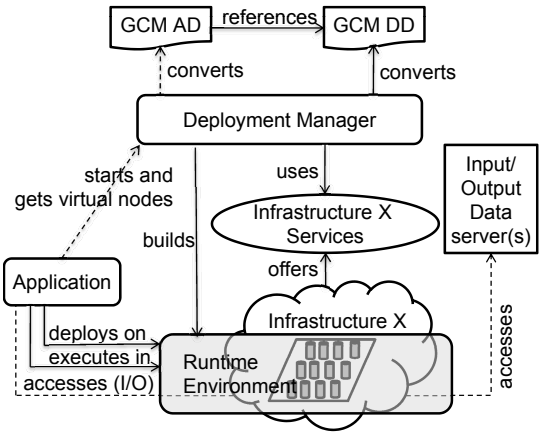

Figure 5. GCM Architecture

A. ETSI Grid Component Model

For users of grid or cloud communities, a provision of common interfaces for the allocation of resources for application deployment in different infrastructures becomes a crucial requirement, since the users wish to access multiple resources of several infrastructures simultaneously and in the most cost saving way. An approach towards such an interface is described in the ETSI GCM standards. The main objective of GCM is the creation of a uniform interface for allocating resources for applications whereas resources may be provided across different grid and cloud infrastructures. The GCM is a gateway approach with a standardized communication protocol based on XML descriptors. The XML descriptors, i.e., in GCM the *Deployment Descriptor* (DD) and the *Application Descriptor* (AD) specify resource information of involved infrastructures in a standardized way.

The content and concepts used in the GCM DD have been derived by abstracting different proprietary interfaces offered by commercial products in the grid, cloud, and cluster computing domains. The key aspect of the GCM specification is the mapping of this abstract interface to different proprietary interfaces of these systems as well as interfaces standardized for this purpose outside of ETSI, e.g., *Open Grid Service Architecture-Basic Execution Service* (OGSA-BES) [14]. Figure 5 shows a generic GCM architecture, which focuses on the GCM AD and DD. It also introduces deployment manager and infrastructure entities to illustrate the likely separation of GCM descriptor processing and provision of the actual resource. Here, the user is assumed to provide a (test) application, a DD XML file, as well as optionally an AD XML file.

The GCM DD describes the resources that can be requested for the deployment of an application on one or more infrastructures. It is converted by the deployment manager into the invocation of specific infrastructure services or commands to reserve resources from the specified infras-

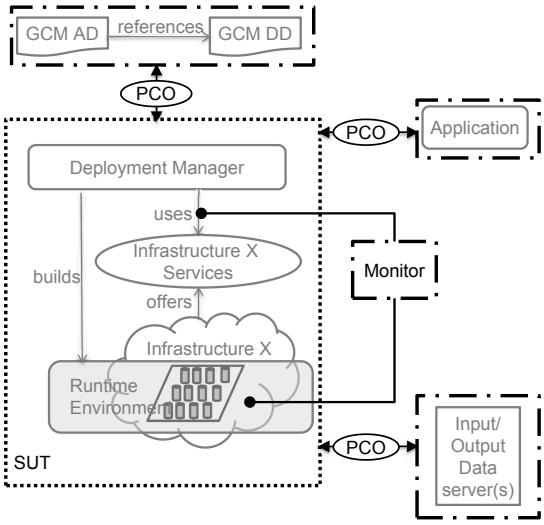

Figure 6. A test architecture for GCM-based deployment

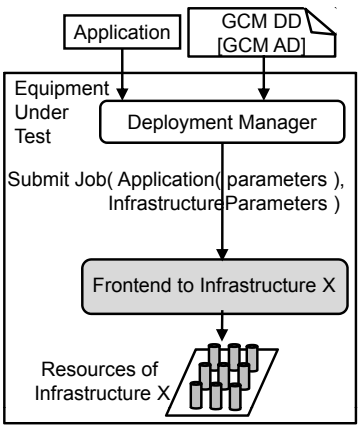

Figure 7. Single infrastructure

tructure(s). The GCM differentiates between infrastructures with direct access to their computing resource - as in the case of a cloud computing system or a set of desktop computers - and indirect access by using a job scheduler - as in the case of a cluster, or a grid middleware. More information and examples of different types of infrastructures can be found in [3].

For the application deployment, a GCM AD specifies the mapping of virtual nodes to real resources as well as the location of input and output data server(s). If a GCM AD is provided, it is used to establish the runtime environment, which is required for application execution.

B. Test architecture

A test architecture for GCM-based application deployment, which follows the concepts defined in [11], [15] is shown in Figure 6. The SUT consists of the deployment manager and at least one infrastructure. The different types of entities that compose the means of testing handle the provision of the GCM DD and AD files to the deployment manager. These entities associated with the infrastructure to be tested evaluate responses from the deployment manager, and analyze the output produced by the application via their *Point of Control and Observation* (PCO). In addition, the processes that run on each infrastructure as well as their interface(s) to the deployment manager and the input/output server(s) are monitored during tests execution. The monitors are *Points of Observation* (PoOs).

The presented testing architecture can also be used to access other standards related to the deployment and execution of applications on grid or cloud infrastructures, e.g., an OGSA-BES web service based interface between the deployment manager and the infrastructure.

C. Test configurations

Test configurations are a refinement of the test architecture. They specify structural aspects of a test and define in detail all equipments participating in a test as well as their communication. Test configurations are then referenced during the specification of tests, which mainly specify behavioral aspects. In this section, we introduce GCM test configurations [16].

1) Single infrastructure: In the test configuration "Single infrastructure", which is depicted in Figure 7, the EUT contains a single infrastructure and the deployment manager. Access to the deployment manager, the infrastructure, the application, the GCM DD, and the GCM AD are available from one single physical machine. The purpose of this test configuration is to keep the complexity low to allow basic testing with minimal effort to establish the test configuration. The user uses the deployment manager to load the GCM DD and in case the test application is a GCM application, also the GCM AD as input. The user is logged locally into the infrastructure to establish the GCM runtime environment and submit jobs related to the application and the infrastructure. If an infrastructure provides indirect access to its resources, e.g., a grid system, a frontend is used to access its resources.

2) Single infrastructure with a bridge: The test configuration "Single infrastructure with a bridge" depicted in Figure 8 has two EUTs, whereas EUT A contains a deployment manager, which is connected via a bridge to EUT B, which contains a single infrastructure. In contrast to the test configuration presented in the previous clause, access to the deployment manager, the infrastructure, the test application to be executed, the GCM DD, and the GCM AD are distributed across two different physical machines. The user is connected remotely to the infrastructure in order to establish the GCM runtime environment and to submit jobs related to the application from the remote machine. This test configuration can be extended with several infrastructures, which are then mapped to EUTs as described in the next

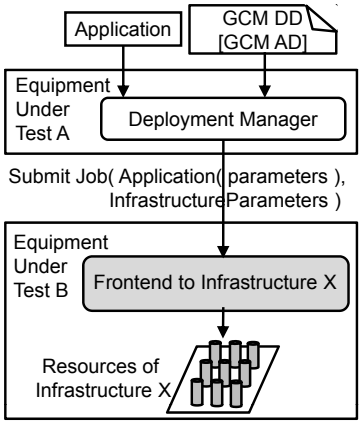

Figure 8. Single infrastructure with a bridge

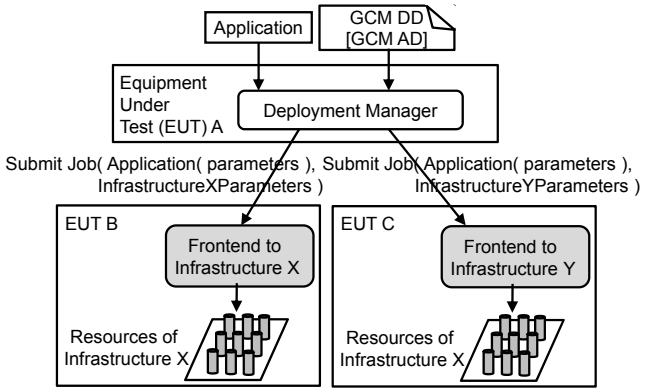

Figure 9. Two infrastructures and bridges

clause.

3) Two infrastructures and bridges: This test configuration is depicted in Figure 9 and extends the test configuration described in the previous clause with a second infrastructure. This test configuration has three EUTs, whereas EUT A contains the deployment manager, EUT B contains the infrastructure X, and EUT C contains the infrastructure Y. Since the deployment manager controls both infrastructures at the same time, it has to be connected to each infrastructure via a bridge.

4) Single infrastructure with a bridge and I/O servers: This test configuration is depicted in Figure 10 and extends the test configuration described in clause V-C2 with input and output data servers. The application can access the input/output data servers from the infrastructure.

D. Compliance Levels

The main purpose of the test framework is the assessment of the standardized GCM AD and DD. The general test objective is to check that applications can be deployed and executed on a given infrastructure based on the information provided in GCM AD and DD. An infrastructure can either provide direct or indirect resource access. To access an

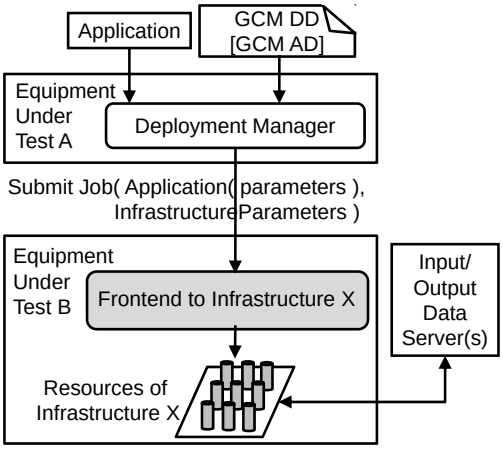

Figure 10. Single infrastructure with a bridge and I/O servers

infrastructure, its protocol need to be followed as specified in the GCM standard [5]. For a classification of functionalities that are provided by a SUT, we define compliance levels as follows:

Compliance by the infrastructure:

1) An infrastructure does not support properties described in GCM AD and DD.
2) An infrastructure supports properties described in GCM AD and DD but are converted in a manual manner.
3) An infrastructure supports properties described in GCM AD and DD and are converted in an automated manner.

Compliance by the deployment manager:

1) Support of multiple infrastructures fulfilling infrastructure compliance level 2.
2) Support of multiple infrastructures where at least one of them fulfills infrastructure compliance level 3 and the others infrastructure compliance level 2 (at least one).
3) Support of multiple infrastructures fulfilling infrastructure compliance level 3.

E. Test purpose specification

The first step in the development of ETSI test specifications is to analyze the base standard and extract testable requirements that are used to specify test purpose. A test purpose specifies if and how a catalogued requirement can be assessed in the context of a given test architecture, i.e., in the form of pre-conditions that are relevant to the requirement, and (a) stimulus and response pair(s). Each test purpose includes at least one reference to the clause in a specification where the requirement to be assessed is described. It should have a unique identifier reflecting its place in the test suite structure.

The GCM standard defines the specification of deployment information and not an interface for the deployment.

TP ID:	TP_GCM_DD_DA_PA_001
Clause Ref:	ETSI TS 102 827 V1.1.1 clause 7.1
Configuration:	Single infrastructure or single infrastructure with a bridge
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD

Figure 11. Test purpose "Single processor with direct resource access"

TP ID:	TP_GCM_AD_VN_001
Clause Ref:	ETSI TS 102 828 V2.1.1 clause 5.2.2
Configuration:	Single infrastructure or single infrastructure with a bridge
Summary:	Ensure that a specific capacity of a virtual node (VN) is enforced as specified in the GCM AD

Figure 12. Test purpose "Specific capacity of a single virtual node"

Therefore, the specification of test purposes for GCM descriptors is not a trivial task. In the case of GCM DD, the primary source of testable requirements is general information associated with resources, such as the number of offered processors or the number of threads per processor available for execution. The secondary source of testable requirements includes parameters that are common to a number of standardized GCM mappings to different infrastructures, e.g., wall time or maximum memory. However, these might be not be supported by each infrastructure. Therefore, a test purpose should not be specific to a single mapping. A third source for additional test purposes includes variations of the requirements mentioned above based on different resource access methods, i.e., direct versus indirect as well as local versus remote access. In the presented testing framework, each test purpose is dedicated to one aspect of a specific requirement or concept defined in the GCM standard.

In Figure 11, an exemplified test purpose for GCM DD is depicted. In this case, the support of the direct resource access is a precondition and a GCM DD with a single processor reservation is the stimulus. The success of the application execution determines the success of the resource reservation.

A test purpose for GCM AD is exemplified in Figure 12. For this test, the support of the GCM AD is required. A GCM AD with a virtual node reservation is the stimulus. The test was successful if the test application is able to allocate the capacity of a virtual node as specified in the GCM AD.

In the development of GCM AD test purposes, (re)assessing of GCM DD information should be avoided. For example, the test purposes for GCM AD should be applicable independently from the method the resources of an infrastructure are accessed (direct or indirect). This means that these test purposes focus on information and concepts specified in the GCM AD. Example source for test purposes is the handling of virtual nodes and input/output data location.

F. Specification of interoperability test descriptions

A test description is a detailed but informal specification of the pre-conditions and test steps needed to cover one or potentially more given test purposes. A test description shall contain the following information:

- **Identifier**: A unique identifier that relates a test to its group and sub-group.
- **Summary**: A unique description of the test purposes covered by this test.
- **Configuration**: A reference to all the equipments required for the execution of this test as well as their connection.
- **Specification References**: One or more references to clauses in the standard for which the test purposes have been specified
- **Test application**: A reference to the test application, which is required to execute this test.
- **Pre-test conditions**: A list of all conditions that have to be fulfilled prior to the execution to a test. These conditions should identify the features that are required to be supported by participating equipment to be able to execute this test, requirements on GCM descriptors, as well as requirements on the parameterization of the test application.
- **Test sequence**: A test sequence is written in terms of external actors and their ability to interact and observe the services provided by the infrastructure, i.e., end-to-end behavior. Based on its success, a test verdict reflecting the interoperability of all EUTs in a test is derived.

The test description can also include a list of checks that should be performed when monitoring the EUT communication on standardized interfaces during the end-to-end test. In the case of GCM testing, this option is not directly relevant since the GCM standard does not intentionally define the interfaces between a deployment manager and infrastructures. However, checks can be formulated if an infrastructure implements interfaces standardized for resource reservation and application execution by other standardization organization, e.g., OGF or *Distributed Management Task Force, Inc.* (DMTF).

An exemplified test description for the GCM DD test purpose shown in Figure 11 is depicted in Figure 13. This test description details a test to check if an infrastructure with direct resource access provides a single processor as specified in the GCM DD. A complete list of the test descriptions can be found in [16].

G. Test applications

For the assessment of the success and validity of each application deployment, a test application is executed on all involved infrastructures. The purpose behind these applications is not to perform complex, real world, computational

Interoperability Test Description		
Identifier:	TD_GCM_DD_DA_PA_001	
Summary:	Ensure that an infrastructure with direct resource access provides a single processor as specified in the GCM DD	
Configuration:	Single Infrastructure or single Infrastructure with a bridge	
Specification References:	GCM DD clause 7.1	
Test Application:	Single process batch job	
Pre-test conditions:	• Infrastructure provides direct resource access • GCM DD contains a direct group description with `hostList` containing one host and host description with `hostCapacity=1` for the infrastructure • Infrastructure has a processor available for use	
Test Sequence:	**Step**	**Description**
	1	User loads the GCM DD and starts the test application on the infrastructure using the deployment manager
	2	Verify that the infrastructure has created and executed the process
	3	Verify that returned application output is correct

Figure 13. Test description "Single processor with direct resource access"

tasks but to produce output that allows determining the real usage of resources and the behavior relevant to a test purpose covered by a test. The test application is parameterizable to allow its reuse across multiple tests.

We determined four different kind of test applications: single process batch job, parallel job, virtual node GCM application, and data manipulation GCM application [16].

The single process batch job starts a single process on a single processor and consumes CPU and memory for a given amount of time. The application's behavior including its execution time, the amount of memory to allocate, and the number of threads can be controlled by parameters. The application prints all information required to determine if a test execution has succeeded or failed either to the standard output or a file. This includes the application start time, the value of each parameter, and the identifier of the application. With this test application, resource deployment and usages can be evaluated.

The parallel job starts a job that uses multiple processes. Each process is mapped to a single processor. The multiple processor application consists of one master process and multiple worker processes. The worker processes communicate with the master process so that the master process receives notifications from all worker processes. A notification should include the host name where the worker process runs and a timestamp. The number of worker processes to be created by the parallel application should be parameterizable. By default, the master process starts up as many worker processes as processors are available, i.e., one node less than specified in the GCM DD. That means that a parallel application requests all available resources. The parallel job prints all the information required to determine if a test execution has succeeded or failed either to the standard output or a file.

The virtual node GCM application starts a deployment as specified in the GCM AD and DD. Once the deployment has been performed, it prints the information provided by each virtual node either to the standard output or a file. For each virtual node, the virtual node name, current number of nodes, and the information about each node used is required.

The data manipulation GCM application starts a deployment as specified in the GCM AD and DD. It deploys a worker on each available node. Each worker reads the same input file from the remote or local input location as specified in the GCM AD. It creates a file with the same content as the input file into the remote or local output location as specified in the GCM AD. Workers should avoid file name conflicts and collisions in the output directory.

H. Test selection and execution

To determine the applicability of a test, all pre-conditions need to be evaluated. To speed up this process, an *Implementation Conformance Statement* (ICS) should be established to allow infrastructure providers to specify supported features prior to a test execution and support automatic test selection. A test should be selected for execution if all of its pre-conditions have been ensured. Common types of pre-conditions in the GCM tests include constraints on:

- the GCM DD and/or AD specifications,
- the infrastructure relating to the type of resource access, features that need to be supported by the EUTs, and available amount of resources,
- and the test application parameterization.

A test should not be selected and recorded as being not applicable if one of its pre-conditions is not met by one (or more) equipment part of the SUT.

A collection and specification of *Protocol Implementation Extra Information for Testing* (PIXIT) can be used to capture infrastructure specific aspects of a GCM DD such as the access details to an infrastructure and resource identifiers, and used to significantly speed up the execution of tests. The developer of an IUT/EUT states the PIXIT that includes information according the IUT/EUT and its testing environment to enable runs of an appropriate test suite against the IUT/EUT [17].

Each grid and cloud infrastructure will be assessed under the same conditions based on a standardized ETSI test specification. Applicable tests are executed by uploading a test specific application, providing infrastructure and deployment information, e.g. via GCM descriptors, and observing the execution of the application as specified in the test specification.

VI. Example test session

This section describes how the tests of the framework can be applied. For this, we apply the test configuration "Two infrastructures and bridges" more specifically as depicted in Figure 14. EUT B contains the grid middleware Globus Toolkit [18] whereas EUT C includes the cloud computing

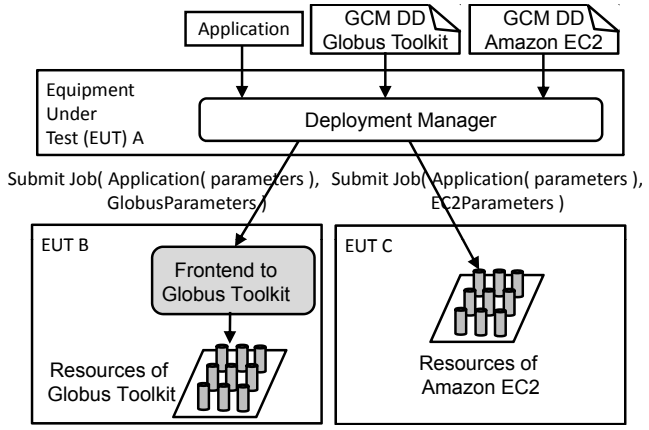

Figure 14. Test configuration "Two infrastructures and bridges" exemplified by Globus Toolkit and Amazon *Elastic Compute Cloud* (EC2)

system Amazon *Elastic Compute Cloud* (EC2) [19], which is an IaaS. For each infrastructure, a GCM DD is needed.

The attributes for describing a DD for Globus Toolkit have already been specified in the GCM standard [5]. The DD used in this test session is depicted in Listing 1. Globus Toolkit is contacted via a *Secure SHell* (SSH)-bridge in order to access the Globus Toolkit frontend. An UNIX-based operating system runs on each computing node whereas each contains four processors, which are represented by the element `hostCapacity` of the attribute `host`. Since Globus Toolkit is an infrastructure with indirect resource access, the total number of available processors is not specified.

A DD for the Amazon EC2 is depicted in Listing 2. A scheme for this infrastructure has not been specified yet, but to experience the application of Amazon EC2 and GCM, this example DD has been developed. In case of its successful deployment, it gives a base for an extension of the GCM standard by the specification of a GCM DD scheme for Amazon EC2. Since Amazon EC2 is an infrastructure with direct resource access, the number of included computing node needs to be specified. In our example, the Amazon EC2 contains ten computing nodes, which are based on a Windows operating system. The infrastructure is accessed via an SSH-bridge. Both presented DDs can be merged into one DD file.

In this test session, we exemplify the test specified in the test description depicted in Figure 15. It will be checked if both infrastructures provide multiple processors for a parallel application. Therefore, the parallel application allocates more than one processor in each infrastructure. The execution of the application will be logged in order to evaluate the result of the test. The Amazon EC2 infrastructure includes ten nodes as described in the DD and the number of nodes in the Globus Toolkit cannot be determined from its DD. Therefore, the parallel test application needs to start ten processes on the Amazon EC2 system and secondly four processes in the Globus Toolkit environment. If all

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <GCMDeployment xmlns="urn:gcm:deployment:1.0"
3    xmlns:xsi="http: // www.w3.org/2001/XMLSchema-instance"
4    xsi:schemaLocation="urn:gcm:deployment:1.0
5      http: // etsi .org/schemas/GCMDDSchemas/extensionSchemas.xsd ">
6    <environment>
7      <javaPropertyVariable name="user.home" />
8    </environment>
9    <resources>
10     <bridge refid="globusGateway" />
11     <group refid="globusGrid">
12       <host refid="ComputeNodeUnix" />
13     </group>
14   </resources>
15   <infrastructure>
16     <hosts>
17       <host id="ComputeNodeUnix" os="unix" hostCapacity="4">
18         <homeDirectory base="root" relpath="${user.home}" />
19       </host>
20     </hosts>
21     <groups>
22       <globusGroup id="globusGrid"
23         hostname="globus.grid.local"
24         bookedNodesAccess="ssh"
25         queue="free">
26         <maxTime>5</maxTime>
27         <stdout>./output</stdout>
28         <stderr>./error</stderr>
29       </globusGroup>
30     </groups>
31     <bridges>
32       <sshBridge id="globusGateway"
33         hostname="grid.informatik.uni-goettingen.de"
34         username="globususer" />
35     </bridges>
36   </infrastructure>
37 </GCMDeployment>
```

Listing 1. GCM DD for Globus Toolkit

```
1  <GCMDeployment>
2    <environment>
3      <javaPropertyVariable name="user.home" />
4    </environment>
5    <resources>
6      <bridge refid="amazonCloudGateway" />
7      <group refid="amazonCloud">
8        <host refid="ComputeNodeWindows" />
9      </group>
10   </resources>
11   <infrastructure>
12     <hosts>
13       <host id="ComputeNodeWindows"
14         os="windows" hostCapacity="1">
15         <homeDirectory base="administrator" relpath="${user.home}" />
16       </host>
17     </hosts>
18     <groups>
19       <amazonCloudGroup id="amazonCloud"
20         hostList="node-[01-10]">
21       </amazonCloudGroup>
22     </groups>
23     <bridges>
24       <sshBridge id="amazonGateway"
25         hostname="aws.amazon.com"
26         username="amazonuser" />
27     </bridges>
28   </infrastructure>
29 </GCMDeployment>
```

Listing 2. GCM DD for Amazon EC2

Interoperability Test Description		
Identifier:	TD_GCM_DD_DA_IA_PA_001	
Summary:	Ensure that an infrastructure with indirect resource access and an infrastructure with direct resource access provide multiple processors for a parallel application as specified in the GCM DD	
Configuration:	Two infrastructures and bridges	
Specification References:	GCM DD clause 7.1, 7.2	
Test Application:	Parallel job	
Pre-test conditions:	• One infrastructures provides indirect resource access • One infrastructures provides direct resource access • GCM DD contains one direct group description and one indirect group descriptions • Communication between the infrastructures is supported • Infrastructures have multiple processors available for use	
Test Sequence:	**Step**	**Description**
	1	User loads the GCM DD and starts the test application on both infrastructures using the deployment manager
	2	Verify that the processes have been created and executed in both infrastructures
	3	Verify that returned application output is correct

Figure 15. Test description "Multiple processors in infrastructures with indirect and direct resource access"

the processes have been started successfully and if the test application writes its output as expected, the test can be evaluated as successful.

VII. Interoperability Event

At ETSI, the validation of interoperability test specifications usually takes place in testing events. Such testing events also provide opportunities for system vendors of the technology under test to assess and demonstrate their interoperability with systems from other vendors. For the GCM testing framework, this validation took place in November 2009 as part of the ETSI Grids, Clouds, and Service Infrastructures event [20]. It provided a unique opportunity for standardization experts, operators, IT service providers, telecom equipment vendors to see available systems running.

A. Application of the Framework in an Interoperability Event

A requirement of the application of the presented testing framework is that a possible SUT needs to include an implementation of the GCM standard. However, the interoperability event included a variety of state-of-the-art systems that implement grid, cloud, cluster, or related technologies, which fit into the idea of the GCM standard. Therefore, as a first step, we compared and executed different state-of-the-art systems to determine and evaluate their similarities and differences. The goal was to feed the result of the demonstrations of the systems of the participated vendors back into the standard to make GCM applicable for these systems. However, this framework is independent of this event and can be applied at other ones as well as in-house.

B. Event Summary

The interoperability event attracted various actors of grid and cloud computing systems from academics, industry, and other standard organizations. In total, six exhibitors demonstrated their grid or cloud environments. The demonstration included resource reservation and application deployment onto different infrastructures. The basis for the evaluation was a questionnaire as well as use case scenarios defined in the ETSI GCM test specification. The questionnaire assessed interfaces for resource reservation and preparation of infrastructure as well as standard support. The use cases included scenarios for infrastructures offering direct and indirect access. Infrastructures were not required to support ETSI GCM standards so that custom-made interfaces were used for application deployment. In addition, capabilities beyond the requirements of the ETSI use cases were shown.

The systems of the vendors who participated in the event mainly implemented a deployment manager for IaaS. However, there was also a deployment manager for PaaS and a cloud management system. All the solutions provided a portal or *Graphical User Interface* (GUI) for resource reservation. For automated use, these have been realized either as RESTful *Web Service* (WS), *Command Line Interface* (CLI), or a Java/XML based API. In general, all the demonstrated systems shield users from complex details of resource reservation. Resource provision and resource requests were handled separately. Handling of data transfer to/from the computing node was mostly done by the application, e.g., using the *Secure Copy Protocol* (SCP) or FTP.

1) Resource Request and Access: Resource request and access were based on different requirements on resources such as computing, storage, or network resources and assessed by the test application referenced in the test description. The resources were selected based on requirements such as performance, *Service Level Agreement* (SLA), application types, or objectives. Fixed or common concepts between the systems could not be identified. The reason may be the application domain specific implementations of the systems. For example, while one system needs a detailed specification of resource requirements, another system only requires a specification of a class defined for resource requirements. In addition, the transparency of the resource management of the systems differed. Therefore, there is a need for an appliance independent hypervisor that manages resources independent of the application.

2) Standard Support: For resource request and access, mainly ETSI GCM, OGF *Distributed Resource Management Application API* (DRMAA), and DMTF *Open Virtualization Format* (OVF) have been implemented. However, most systems use non compliant default configuration, but also allow adaptation to DMTF OVF.

A few basic standards are supported by commercial cloud systems, since cloud computing is an emerging technology and standards are only slowly evolving. Most of the cloud systems provide proprietary RESTful WS and XML based interfaces to resources. This provides a simple basis for further standardization and extensions.

Weak points of existing standards are that they allow too many options such as in the OGF *Job Submission Description Language* (JSDL) or that they require to fix the location of resources. Most desired is a standardized API for virtual machine and resource management.

3) Test automation: The presented testing framework has been mainly developed for the application in interoperability events. The tested systems can be seen as black boxes connected and accessible only by their interfaces. Automating test executions in such a configuration is challenging since agreed standards are not available for the usage of grid or could infrastructures. Furthermore, in the setting of such events participants are usually known as late as two weeks before the event. However, test specification efforts usually are concluded months before an event. Therefore, and due to time and budget limitations, test execution cannot be automated by the organizer. For participants, this would be an extra cost they are not willing to spend. Therefore, the tests have been conducted manually.

4) Reflections: Key areas for standardization, i.e., clear boundaries of the state-of-the-art systems is an API for the provision of resources and for requests of resources. Open issues are the achievement of portable appliances of the hypervisor, i.e., the management of different virtual machine images and resources. Minor concerns include lack in agreed terminology and the need for a strong common denominator. Cloud standardization needs are considered in detailed in the following section.

C. Identified Needs in Cloud Standardization

In the conducted interoperability event, we identified several standardization needs for cloud computing infrastructures related to interoperability. These needs are related to the forms of cloud computing as described in Section II.

For IaaS clouds, functionalities such as the management of resources, application, and data but also common security (authentication and authorization), billing, and accounting interfaces need to be standardized. A cloud resource management standard should consider interfaces for the deployment of virtual machines including their start, stop, status requests, image format, monitoring, performance measurement, and access. Similar to the resource management, cloud application should be management in a common way. This includes their deployment, start, stop, and status. Cloud data management includes especially their access.

On the PaaS level, the platform uses the standardized interfaces described above. Such cloud platforms should offer further features such as dynamic resource allocation and abstraction of resources through a standardized API. In addition, it should be possible to import and use other PaaS interfaces. The SaaS is then implemented using such a standardized cloud platform API.

According to interoperable grid and cloud infrastructures, an application should be able to use them simultaneously. For this, commonly agreed protocols are required to exchange information and to allow their management. A result would be a cloud/grid broker, which the user accesses to use functionalities of grid and cloud systems. Further consideration on cloud standardization requirement can be found in [21].

VIII. Related work

ETSI developed and published a methodology for interoperability testing [11] and automated interoperability testing [15]. In the latter, guidelines and best practices for automated interoperability testing are presented. This methodology has been applied successfully for the development of interoperability test specifications for various technologies, e.g., IPv6 [22] and *Internet Protocol* (IP) *Multimedia Subsystem* (IMS) [23], and put into practice in ETSI interoperability events [13]. Previously, this approach had not been applied to grid or cloud computing technology.

Several interoperability and standard initiatives for grid and cloud computing systems exit. For cloud systems, these include the OGF *Open Cloud Computing Interface* (OCCI) Working Group [24], the IEEE Standards Association [25], the DMTF *Cloud Management Standards* [26], and the *Open Cloud Consortium* (OCC) [27]. The activities of major cloud standardization initiatives have been summarized in a report by the *International Telecommunication Union* (ITU) [28]. These standardization activities are diverse and each initiative chooses the flavors of cloud computing that fit best to their requirements. This is one reason why the concepts of cloud computing are not fully agreed on. However, no interoperability test specifications for grid and cloud computing systems have been published, yet.

Bernstein et. al. identified areas and technologies of protocols and formats that need to be standardized to allow cloud interoperability [29]. They call this set of protocols and formats Intercloud protocols because they should allow cloud computing interoperability. If this set of protocols will be commonly accepted, the GCM and the interoperability testing framework presented in this paper could be adapted to improve cloud interoperability.

Merzky et. al. present application level interoperability between clouds and grids based on SAGA, a high-level interface for distributed application development [30]. The interoperability is achieved by cloud adapters. These adapters are specific to the Amazon Cloud and the Globus Toolkit.

Interoperability initiatives such as OGF *Grid Interoperability Now* (GIN) and standards bodies in grid computing are described in [4]. OGF interoperability test specifications for grid are rarely available and only for selected standards such as GridFTP. Also, they follow rather ETSI's notion of conformance testing than interoperability testing. Due to our knowledge, an interoperability testing framework for such

diverse domain of grid and cloud computing infrastructures has not been published.

IX. Summary and Future Research

In this paper, we discussed differences between grid and cloud computing infrastructures. Furthermore, we presented generic approaches on achieving interoperability of distributed systems and different types of testing including interoperability testing with conformance checking.

Our main contribution is a testing framework around the GCM standard developed by ETSI that is applicable for grid, cloud, and cluster management systems. This framework has been developed by following the ETSI test development process. We described the GCM architecture, an applicable test architecture, developed test configurations, and test applications. Test purpose specification and interoperability test descriptions have been explained. Furthermore, we considered test selection and test execution according to the presented testing framework. The application of the framework has been exemplified by Globus Toolkit and Amazon EC2 as EUTs. As a result from the application of the framework in the ETSI Grids, Clouds, and Service Infrastructures event, we identified needs toward standardized grid and cloud infrastructures.

We believe that the GCM standard is a first step towards the use of resources from different infrastructures simultaneously in a standardized way. Infrastructure adapters and translators can easily be incorporated into the standard. With the presented testing framework, it is possible to enhance and expand the standard to allow a wide adaption of several systems provided by different vendors.

The described testing framework is part of an initiative for standardizing the use of grid and cloud technology in the context of telecommunication at ETSI. We believe that this testing framework is a step towards systematic interoperability testing of grid and cloud computing environments in general [3].

The specification of executable test cases from GCM test descriptions is considered as future work. This step includes the further concretization of GCM test configurations including equipment user and monitor test components as well as the specification of their behavior. We consider to automate the tests following the methodology on automated interoperability testing [15], which is a challenging task. Furthermore, we plan to extend the testing framework to make it applicable for upcoming cloud standards.

Acknowledgment

The work carried out here is co-financed by the EC/EFTA in response to the ECs ICT Standardisation Work Programme.

References

[1] ETSI Technical Committee CLOUD (TC CLOUD), previously TC GRID, [Online; http://portal.etsi.org/cloud fetched on 10-06-11].

[2] "ETSI TR 102 659: GRID;Study of ICT Grid interoperability gaps," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

[3] "ETSI TR 102 766: GRID; ICT Grid Interoperability Testing Framework and survey of existing ICT Grid interoperability solutions," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

[4] T. Rings, G. Caryer, J. Gallop, J. Grabowski, T. Kovacikova, S. Schulz, and I. Stokes-Rees, "Grid and Cloud Computing: Opportunities for Integration with the Next Generation Network," *Journal of Grid Computing: Special Issue on Grid Interoperability, JOGC*, vol. 7, no. 3, pp. 375 – 393, 2009.

[5] "ETSI TS 102 827: GRID;Grid Component Model (GCM);GCM Interoperability Deployment," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.

[6] "ETSI TS 102 828: GRID;Grid Component Model (GCM);GCM Application Description," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2010.

[7] "ETSI TS 102 829: GRID;Grid Component Model (GCM);GCM Fractal Architecture Description Language (ADL)," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

[8] "ETSI TR 102 767: GRID;Grid Services and Telecom Networks;Architectural Options," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

[9] T. Rings, J. Grabowski, and S. Schulz., "On the Standardization of a Testing Framework for Application Deployment on Grid and Cloud Infrastructures," in *Proceedings of the 2nd International Conference on Advances in System Testing and Validation Lifecycle (VALID 2010)*. IEEE Computer Society, 2010, pp. 99–107.

[10] H. van der Veer and A. Wiles, "Achieving Technical Interoperability - the ETSI Approach," White Paper, European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.

[11] "ETSI ES 202 237: Methods for Testing and Specification (MTS);Internet Protocol Testing (IPT);Generic approach to interoperability testing," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2007.

[12] ETSI, "ETSI Standard (ES) 201 873 V3.2.1: The Testing and Test Control Notation version 3; Parts 1-8," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, also published as ITU-T Recommendation series Z.140, 2007.

[13] ETSI, "Plugtests™Interop Events," [Online; http://www.etsi.com/WebSite/OurServices/plugtests/home.aspx fetched on 10-06-11].

[14] I. Foster, A. Grimshaw, P. Lane, W. Lee, M. Morgan, S. Newhouse, S. Pickles, D. Pulsipher, C. Smith, and M. Theimer, "OGSA Basic Execution Service Version 1.0, GFD-R.108," Open Grid Forum, 2008.

[15] "ETSI EG 202 810: Methods for Testing and Specification (MTS);Automated Interoperability Testing;Methodology and Framework," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2010.

[16] "ETSI TS 102 811: GRID;Grid Component Model (GCM);Interoperability test specification," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2010.

[17] ISO/IEC, "Information Technology – Open Systems Interconnection – Conformance testing methodology and framework," International ISO/IEC multipart standard No. 9646, 1994-1997.

[18] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems," in *Proceedings of the IFIP International Conference on Network and Parallel Computing (NPC05)*, ser. LNCS, vol. 3779. Springer, 2005.

[19] "Amazon Elastic Compute Cloud (Amazon EC2)," [Online; http://aws.amazon.com/ec2/ fetched on 10-06-11].

[20] ETSI, "Grids, Clouds & Service Infrastructures: Plugtests™ and Workshop," [Online; http://www.etsi.com/plugtests/GRID09/GRID.htm fetched on 10-06-11].

[21] "ETSI TR 102 997: CLOUD;Initial analysis of standardization requirements for Cloud services," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2010.

[22] "ETSI TS 102 517: Methods for Testing and Specification (MTS);Internet Protocol Testing (IPT): IPv6 Core Protocol;Interoperability Test Suite (ITS)," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2008.

[23] "ETSI TS 186 011-2: Technical Committee for IMS Network Testing (INT);IMS NNI Interworking Test Specifications;Part 2: Test descriptions for IMS NNI Interworking," European Telecommunications Standards Institute (ETSI), Sophia-Antipolis, France, 2009.

[24] OGF, "Open Cloud Computing Interface Working Group," [Online; http://forge.ogf.org/sf/projects/occi-wg fetched on 10-06-11].

[25] IEEE Standards Association, "P2301 - Guide for Cloud Portability and Interoperability Profiles (CPIP)," [Online; http://standards.ieee.org/develop/project/2301.html fetched on 10-06-11].

[26] DMTF, "Cloud Management Standards," [Online; http://www.dmtf.org/standards/cloud fetched on 10-06-11].

[27] "Open Cloud Consortium," [Online; http://opencloudconsortium.org/ fetched on 10-06-11].

[28] ITU Telecommunication Standardization Bureau, "Activities in Cloud Computing Standardization - Repository," 2010, [Online; http://www.itu.int/dms_pub/itu-t/oth/49/01/T49010000020002PDFE.pdf fetched on 10-16-11].

[29] D. Bernstein, E. Ludvigson, K. Sankar, S. Diamond, and M. Morrow, "Blueprint for the Intercloud - Protocols and Formats for Cloud Computing Interoperability," in *ICIW*, M. Perry, H. Sasaki, M. Ehmann, G. O. Bellot, and O. Dini, Eds. IEEE Computer Society, 2009, pp. 328–336.

[30] A. Merzky, K. Stamou, and S. Jha, "Application Level Interoperability between Clouds and Grids," in *GPC Workshops*. IEEE Computer Society, 2009, pp. 143–150.

Adaptive Video Streaming through Estimation of Subjective Video Quality

Wolfgang Leister, Svetlana Boudko, and Till Halbach Røssvoll
Norsk Regnesentral
Oslo, Norway
Email: {wolfgang.leister, svetlana.boudko, till.halbach.rossvoll}@nr.no

***Abstract*—Concerning video transmission on the Internet, we present a model for estimating the subjective quality from objective measurements at the transmission receivers and on the network. The model reflects the quality degradation subject to parameters like packet loss ratio and bit rate, and is calibrated using the prerecorded results from subjective quality assessments. Besides the model and the calibration, the main achievement of this paper is the model's validation by implementation in a monitoring tool. It can be used by content and network providers to swiftly localise the causes of a poor quality of experience (QoE). It also can help content providers make decisions regarding the adjustment of vital parameters, such as encoding bit rate and error correction mechanisms. We show how the estimated subjective service quality can be applied for decision making in content delivery networks that consist of overlay networks and multi-access networks.**

***Keywords*—Quality of Experience; perceived quality; video streaming; assessment; adaptation**

I. Introduction

Streaming of video content to a broad public is a well-known technology and increasingly used both in stationary and mobile applications. The technical quality of such streams is essential for the viewers. Previously, we presented the estimation of subjective video quality as feedback to the Content Provider (CP) [1]. We extend the metrics used there to control an overlay network for video streaming including a mobile scenario.

In a stationary scenario, content from the CP is sent through the networks of an Internet Service Provider (ISP) to the consumer's home network and can then be viewed on, for instance, a TV screen. While this provides an effective way of distributing streamed content, the CP may not be able to serve all consumers due to capacity limitations of the network. Therefore, CPs position streaming servers at particular nodes with preferred ISPs from which the content is conveyed to the consumers.

CPs may experience that consumers complain about a reduced image quality even though the culprits are problems at the ISP's or the consumer's location. This includes systems and home network on the consumer side. Also bandwidth sharing with other devices and the use of wireless networking devices are known to be problematic. In the MOVIS project [2], which forms the motivation of this paper, a monitoring system that collects objective data from several sources and estimates the assumed consumers' satisfaction has been implemented. The tool integrates the data and provides an analysis to the customer service regarding the possible causes of a problem.

The metrics for the assumed customer satisfaction are not limited to measurements and alerts, but can also be used to control the content delivery network.

For the metrics for measurements and alerts, we analyse the delivery chain for video on demand and identify a number of factors that influence the quality as experienced by the consumer using the scenario sketched in Figure 1. Multimedia content is streamed from the CPs, routed through the network of ISPs, and is finally accessed by consumers, typically in private homes with broadband access. The perceived quality of experience (QoE) for the consumer can be affected at all stages in the delivery process. For the consumer, the overall quality of the video stream is most important in a pay-per-service model.

In the following sections, we show how to estimate the QoE for single consumers and groups of consumers given the above scenario. In Section II, we give an overview of models and metrics for measuring the quality of service (QoS) and estimating the QoE for streamed content. In Section III, we propose a novel model for estimating the QoE, and we show results from an assessment process for video content in Section IV. How to control a content delivery network using an overlay network and a multi-access network is shown in Section V. We conclude in Section VI, showing how our model is used in practice.

II. Estimation Models and Metrics

The user-perceived quality of a service is affected by numerous factors in the end-to-end delivery path. For measuring the QoS, there are several approaches that can be classified by whether they are subjective or objective, direct or indirect, in-service or out-of-service, real-time or deferred time, continuous or sampled, intrusive or non-intrusive, and single-ended or double-ended [3].

Existing methods for QoS measurement can be classified into network and application level measurements. Examples of widely used metrics for network-level QoS include connectivity (RFC 2678); one and two-way delay (e.g., RFC 2679, RFC 2680, RFC 2681); one-way delay variation (jitter; see the IP Performance working group IPPM); throughput; and packet

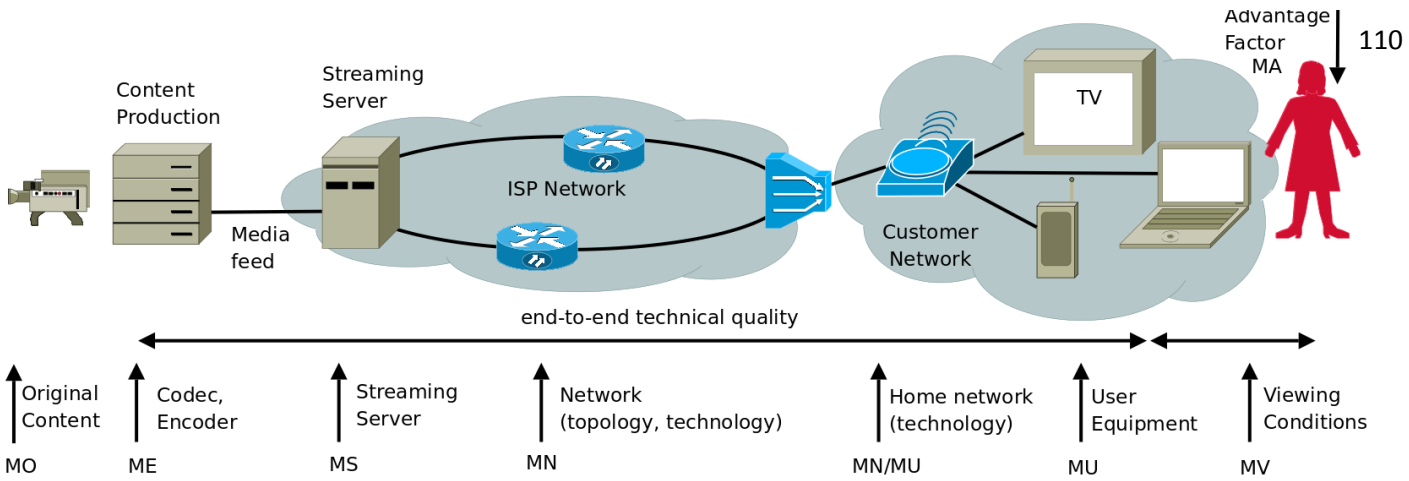

Figure 1. Transmission chain. Entities which have an influence on the consumer's experience of quality are identified.

loss, as described in the framework for IP-based metrics in RFC 2330 [4], and WOAMP (active measurement protocols.

Observations of QoS on all levels can be made by non-intrusive measurement (passive observation of QoS at the end system), or intrusive measurement (the controlled injection of content into the stream in order to make deductions of the quality). These measurements are performed according to a QoS metric that quantifies the performance and reliability of the Internet [4]–[6].

For measuring the QoS, there are several approaches that can be classified based on whether they are subjective or objective, direct or indirect, in-service or out-of-service, real-time or deferred time, continuous or sampled, intrusive or non-intrusive, and single-ended or double-ended.

Existing work that claims to derive QoE values directly from network QoS frequently applies peak-signal-to-noise ratio (PSNR) as its metric for translating QoS to QoE. There are very strong arguments against this practice, for example [7], [8]. The picture appraisal rating for MPEG-2 (PAR) [9] assesses the quality of indivual pictures instead of videos as well and suffers the same problems as PSNR. A very simple approximation in this general family of approaches was proposed [10]. Rate-distortion models as introduced [11] are also based on PSNR, which constitutes a problem, but are less affected by unstable quality because quality is adapted by changing quantization factors over time.

The just noticable differences (JND) provides an objective metric that tries to emulate the human visual apparatus in a way that considers the change of pictures over time, as do approaches like the Video Quality Metric (VQM) [12]. The VQM uses a double-ended approach that can be adapted for continuous in-service quality monitoring when using an additional ancillary data channel. VQM is based on the extraction of perception-based features and the computation of particular quality parameters.

Koumaras et al. [10] presented a theoretical framework for end-to-end video quality prediction. This framework is independent of the video codec, and the dynamics of the encoded sequence. It consists of two models. The first model operates at the pre-encoding stage and predicts the video quality of the encoded signal. Here, the authors showed that the dependency between the bit rate and the perceived quality of service is described by a logarithmic function $\langle \mathrm{PQoS} \rangle_{\mathrm{SSIM}} = 0.1033 \ln(x) + 0.2940$. Their second model maps the packet loss rate to the video quality degradation, whereby the quality of the transmitted video is linearly dependent on the percentage of the successfully decoded frames. We realize that the estimation of perceived video quality is an open research field. Video quality experts agree that existing quality estimation methods must be used carefully [13].

User perceived quality metrics typically take the characteristics of the transported content and network level QoS into account. Traditionally, the QoE has been assessed by means of user evaluations. We found also tools that estimate the QoE from objective measurements at the application level.

Subjective quality measurement involves a test panel with individuals, while the objective measurements are performed on the media content [14], [15]. In voice communications and image processing, the mean opinion score (MOS) provides a numerical measure for the QoE using subjective tests (opinionated scores) that are statistically averaged to obtain a quantitative indicator of the system performance. Pre-recorded samples are played back to a mixed group of people under controlled conditions, using the rating: (1) bad; (2) poor; (3) fair; (4) good; (5) excellent. The MOS is the arithmetic mean of all the individual scores.

Standards for performing user assessment studies to determine QoE for TV and multimedia can be found in DSCQS / BT.500-11 [16] and SAMVIQ / BT.700 [17]–[19]. Applicable standards for subjective video quality assessment by the EBU include ETR 290, TR 101 290 (Measurement guidelines for DVB systems), and TR 101291. SAMVIQ builds on the experiences of the ITU standard BT.500-11 for video, on experiences from the audio assessment methodology MUSHRA, and

on the ITU-T recommendation P.911 (Subjective Audiovisual Quality Assessment Methods for Multimedia Applications).

Examples of methodologies defining the criteria of objective and subjective quality for video compression include the work of Bennett and Bock [20], where a signal noise analysis to compare compatible MPEG and WM9 codecs is presented. As video streaming involves more than solely video compression and is subject to the quality of network protocols, physical lines, software media players, computer hardware and other components, the impact of these components has been studied. Video streaming technologies with user tests to evaluate the perceived quality have been analysed [21], also taking the effects of quality adaptation methods such as H.264 SVC or frequent layer switching into account [22], [23].

For audio, several methods and sophisticated algorithms have been developed to evaluate the perceived QoS, such as the E-model (ITU G.107), the R-value [24], PSQM (ITU P.861), PAMS (BT), PEAQ (ITU-R BS.1387), PESQ [25], [26], and the single ended objective measurement algorithm in P.563 [27].

The E-model of the ITU-T Recommendation G.107 [28] uses transmission impairment factors based on a concept given in the description of the so-called OPINE model. The result of a calculation with the E-model is the transmission rating factor R, which is composed of $R = R_0 - I_s - I_d - I_{e-eff} + A$ where all of the factors are composed of several sub-factors. R_0 represents the basic signal-to-noise ratio, including noise sources such as circuit noise and room noise. The factor I_s represents the impairments occurring simultaneously with the voice transmission; I_d describes all impairments due to the delay of voice signals, while I_e describes the equipment impairment factor, which is derived from a set of values described in ITU-T Recommendation G.113. The advantage factor A describes psychological aspects under the viewing.

The Application Performance Index (APDEX) [29], [30] is a numerical measure of consumer satisfaction for groups of consumers, for instance for measuring response times. User ratings are categorised into *satisfied*, *tolerating*, and *frustrated*. APDEX_T is then calculated from the numbers in each category, and ranked into five quality classes from *unacceptable* to *excellent*.

III. Quality degradation model

In the scenario shown in Figure 1, the streamed data are transported from the CP through various networks to the device where the stream is presented. Each entity along this chain can potentially mean a decrease in quality as compared to the original quality Q_O. We account for this by using the model defined below that estimates the subjective QoE both for single consumers based on factors reducing the original quality of content, and for groups of consumers based on the APDEX formula discussed above.

A. QoE for one consumer

Inspired by the E-model, we estimate the quality perceived by the end user as a product of the original quality and a number of influencing factors. Each factor is related to a certain entity of the delivery chain in our scenario and hereby represents the respective impact on the consumer quality. Thus, the estimated QoE for one consumer is defined as

$$\tilde{Q} = Q_\text{O} \cdot \prod_{i \in \{\text{E,S,N,U,V,A}\}} M_i,$$

where Q_O is the original quality measure, $M_\text{A} \geq 1$, and $0 < M_i \leq 1$ for $i \in \{\text{E,S,N,U,V}\}$. Setting $M_i = 1$ denotes the lack of influence, such as a transparent channel. In the following we describe each single factor:

- M_E: Influence of the encoding on the delivered content. It depends on the codec, codec parameters, and the content (fast vs. slow movements, colour, contrast, etc.).
- M_S: Influence of the streaming server on the delivered content. It depends on the streaming protocol, the implementation of the streaming server and, to a certain extent, on the codec.
- M_N: Influence of the network on the delivered content. This factor is influenced by technical parameters like delay, jitter, congestion, packet loss, and out of order packet arrival. It also depends on the used codecs and protocols. The parameter consists of three distinct parts: the CP network, the ISP network, and the consumer network.
  Possible influences from the hardware at the consumer's home (e.g., routers, WLAN, sharing with other devices) are taken into consideration.
- M_U: Influence of the consumer's equipment on the delivered content. Hardware type and parameters (e.g., CPU speed, memory size), system and application software, and a system load parameters have an influence on M_U.
- M_V: Influence of the viewing conditions in the consumer's home, such as distance to the viewing screen.
- M_A: Advantage factor from the use of the content, modelling cognitive effects like the acceptance of a grainy image sometimes encountered with older films. This increases the value of the content subjectively, even if the technical QoE is worse.

Note that in the general case, some of the factors are not orthogonal, i.e., they depend on the impact factors of previous steps. For instance, a particular networking error can be visible in different manners for different codecs or bandwidth settings.

The factors M_i must be derived from an objective measurement, here called assessment process, and mapped/scaled to the allowable range of values as defined above. Ideally, a model calibration process using regression analysis could be used to derive the scale factors for calculating $\tilde{Q}$. However, this requires a large data set of measurements, so that the dependencies between all input parameters can be derived. We cannot present such a data set, since this would require to perform a very large number of SAMVIQ assessments, which are rather costly and time consuming. Instead, we simplify

the model and select only significant parameters, as detailed below.

B. Simplified model and parameter scaling

As M_{U} is outside the control of CPs and ISPs, we assume $M_{\mathrm{U}} = 1$ in this work. Notably, we argue that most modern media playing and viewing devices have enough computing power to decode video streams, and that there is not too much additional load on the processing unit. We assume memory to be available in a sufficient amount. Next, even though the M_{V} is usually considered in assessment processes by applying approved standard settings (for instance specified in SAMVIQ), we set $M_{\mathrm{V}} = 1$ for the sake of simplicity, assuming perfect viewing conditions. Finally, $M_{\mathrm{A}} = 1$ as the advantage factor obviously is outside the scope of this paper.

M_{E} and M_{S} depend on encoding and streaming parameters. Because they are controlled by CPs and ISPs, both factors can be combined as $M_{\mathrm{E,S}}$, which is determined as follows. In a representative pre-evaluation, several videos covering various content characteristics like sports or talking head are encoded with varying parameters and streamed. The subjective quality of the decoded frames is recorded as a function of the most important parameter, the bit rate allocated to the sequence. Given a particular bit rate and content type, the respective score can be found by means of a function derived from pre-recorded measurement points. The score is then scaled linearly to the range (0,1] and hereby becomes identical with $M_{\mathrm{E,S}}$. We show how to derive this factor in Section IV-B.

M_{N} depends on the networks of providers and consumer, which results in measurable delay, jitter, and packet loss. For the QoE of the consumer, the resulting packet loss is most important. Modern media players typically employ some form of buffers, such that jitter and delay eventually will result in packets arriving too late for decoding, and thus contribute to packet loss. Therefore, we consider packet loss as the relevant parameter for M_{N}. We derive M_{N} from an assessment that evaluates the influence of packet loss on the QoE. Under varying network conditions, the influence of packet losses on the perceived QoE is measured, and the resulting score is scaled to fit the range (0,1]. Again, this pre-evaluation is made available to the CPs and ISPs in the form of look-up tables. Note that the values measured also depend on encoding parameters.

Packet loss is the result of bit errors that cannot be corrected, bursty packet loss occurring for other reasons, delay above the buffer size, and jitter above a certain threshold. Note that these are related to each other. Typical values are: $\sim 30-40$ ms for delay [31], ~ 14 ms for jitter [31], and ~ 5 ‰ for packet loss [32], concerning wired ADSL connections. These values are too small to show a significant impact on the QoE, as shown in Section IV-C. However, when the consumer uses a WLAN, the packet loss rate is typically around 5% [33]. Also, in the case of network congestion, significantly higher values apply, e.g., a packet loss rate above 10% [33]. Finally, bursty packet loss appears when wireless networks are used, which can cause different QoE degradation than a randomly distributed packet loss pattern.

C. QoE for groups of consumers

While $\tilde{Q}$ describes the QoE for one consumer, the ISPs need a measure to describe the QoE of a group of consumers who share common resources, such as a common router or DSLAM. For this we apply the APDEX model [30] as follows. We classify the consumers into three quality classes according to their current $\tilde{Q}$, given the threshold values T_{S} and T_{U}, like so: The consumer is in the set $M^{(\mathrm{S})}$ for $\tilde{Q} > T_{\mathrm{s}}$, in the set $M^{(\mathrm{T})}$ for $T_{\mathrm{s}} > \tilde{Q} > T_{\mathrm{u}}$, and in the set $M^{(\mathrm{U})}$ else. The threshold T_{S} is suggested to be at 60-80% of the maximum $\tilde{Q}$, and T_{U} at about 40%, depending on the expectations of the consumers. We then apply the formula

$$A_{\mathrm{M}} = \frac{\left|M^{(\mathrm{S})}\right| + \left(\left|M^{(\mathrm{T})}\right|/2\right)}{|M^{(\mathrm{S})}| + |M^{(\mathrm{T})}| + |M^{(\mathrm{U})}|},$$

and rank A_{M} into U (unacceptable), P (poor), F (fair), G (good) and E (excellent) with the threshold values $0 \leq \{U\} \leq 0.5 < \{P\} \leq 0.7 < \{F\} \leq 0.85 < \{G\} \leq 0.94 < \{E\} \leq 1$. These quality classes are visualised, e.g., using a colour code, for groups of consumers.

IV. Subjective Quality Assessment

In order to establish the correlation between known parameters, such as bit rate, content type, and packet loss, and estimated subjective quality, in particular M_{E}, M_{S}, and M_{N}, our project partner *Institut für Rundfunktechnik* (IRT) conducted assessments for the WM9 codec [34], [35] using the SAMVIQ (Subjective Assessment Methodology for VIdeo Quality) method. The choice of settings and parameters used in the assessment, such as encoding parameters, frame rate, or format, is guided by the practical needs of the CP participating in our project.

A. Assumptions and methodology

For the assessment we use the SAMVIQ method as a tool, rather than developing an own assessment method. SAMVIQ is well-known, and defines experiment settings, the workflow necessary for an assessment, and how to ensure a statistically significant result. In the assessment different qualities are rated without the subjects knowing what they are rating. While this method provides a metric for measuring quality, it is unaware of any technical contexts, such as encoding parameters or measured objective values.

While we can observe how varying parameters have an impact on the perceived quality, the method cannot decide on the cause for a particular results. In case of an unexpected assessment outcome, other methods need to be employed to find out the specific causes, such as inspecting the code of the software or analysing the impact of protocol parameters.

B. Assessment of encoding and streaming parameters

How the configuration of the encoding and streaming tiers influence the perceived quality is discussed below.

For practicality purposes, we define the production quality factor $M_P = M_E M_S$ as the product of the encoding and streaming quality factors, as M_E and M_S are partly correlated due to a joint dependency on the used codec. M_P depends on a number of factors. Scene type, spatial resolution, and bit rate varied in our experiments, as detailed below. Other parameters were set to a reasonable value to keep the complexity at a moderate level.

Four image sequences with different content were used. "Skiing" is a bright, almost monochromatic sequence with little camera motion. "Rugby" is moderately detailed, with a lot of camera motion and moving objects. "Rainman" is highly detailed but contains only little camera motion, while "Barcelona" is very colourful and detailed. All sequences were originally in SD quality. CIF resolution videos (352×288 pixels) were produced from SD material by means of cropping and subsampling. The duration of the sequences was chosen to be roughly 10 s in order for a sequence to contain a uniform scene type. However, since 10 s is too short to achieve a stationary bit rate control, each 10-second sequence is repeatedly fed into the encoder, resulting in — with 4 repetitions — a total duration of 40–60 s. However, only the last sequence was actually shown during the visual assessment. The encoder was configured to operate with a fixed given bit rate, ranging from 80 Kbps to over 1 Mbps with CIF size image material, and from 800 Kbps to 10 Mbps with SD imagery.

The frame rate of the SD sequences was 25 Hz, whereas the frame rate of the CIF sequences varied from 6.25 fps at 80 and 168 Kbps to 12.5 fps at 352 Kbps and 25 fps for higher bit rates, in order to allow for a minimum image quality at very low rates. Below a rate of 384 Hz, the encoder was operated in Simple Profile (Medium Level and Low Level, depending on the bit rate), while the Advanced Profile, Level 0 was turned on at higher bit rates. The key frame distance was set to 2–3 s, to balance the bit rate consumption of intra frames, which demands as few intra frames as possible, against (re-)synchronisation ability constraints, which calls for frequent insertion of intra frames, as intra frames can be used by decoders to regain synchronisation with the bit stream in case of transmission errors or when switching channels. Concerning the encoder's quality parameter, the best compromise between smoothness and sharpness were yielded at a value of 50. The final parameter to adjust, the buffer size, was set to contain an entire group of pictures, i.e., all frames in the interval from one intra frame to the other.

The subjective assessment follows the SAMVIQ method introduced above. Eighteen individuals participated in the testing in total. All met identical viewing conditions and started the testing session with a training phase where they, after having received instruction, could become acquainted with their task. During the testing, the sequences encoded with differing bit rates were presented to the participant in random order, with the original sequence in the first, and the tester was then asked to assign a proper score for the perceived image quality, ranging from 0 to 100 (excellent). Comparisons with other scores for other sequences and adjustment of all scores simultaneously was possible. The original sequence was hidden among the test sequences as a means to control each personal assessment, and to sort out statistical outliers.

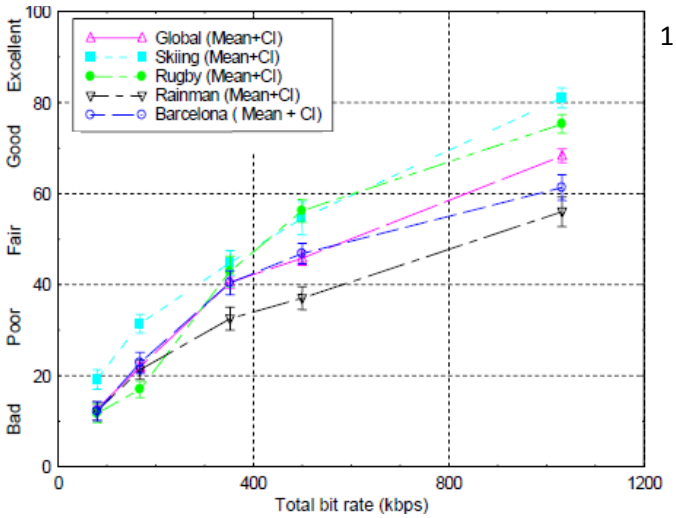

Figure 2. Recorded subjective quality assessment rate using the SAMVIQ scale as a function of content and encoding rate. WM9 codec, CIF video format.

The evaluation's outcome is as follows. There were three outliers. Those data were discarded, leaving fifteen valid participant contributions. The data have been aggregated in rate distortion curves, where the distortion/quality is measured in the SAMVIQ scale that directly translates to the shown Mean Opinion Square (MOS). When normalised by the maximum scale, the values correspond to the production quality factor M_P. Figure 2 summarises the assessment for all sequences with CIF resolution, including the average ("Global"), while Figure 3 shows the average for SD image material.

As can be seen, the average CIF curve is nearly logarithmic. Nearly 400 Kbps are needed to achieve a "Fair" quality assessment. For a "Good" quality or better, at least 800 Kbps are necessary. A bright sequence like "Skiing" and one with a moderate amount of detail like "Rugby" can easier meet a "Fair" or "Good" quality constraint. In contrast to that, "Rainman" with a lot of high-frequency content and the highly detailed "Barcelona" need 1100 Kbps and 1000 Kbps, respectively. Such a spread favours the taxonomy of various image sequences according to a few key parameters over a simplification with a single average, as the variance of bit rates to achieve for instance a "Good" quality is very large. Examples for key parameters are amount of detail, amount of colour, number of moving objects, camera movement (including zoom, turn, etc.), and number of scene changes.

The sequence average curve for SD-sized imagery shows that "Fair" is achieved at a rate of roughly 2 Mbps, "Good" at 3 Mbps, and "Excellent" at 5 Mbps. In general, this curve

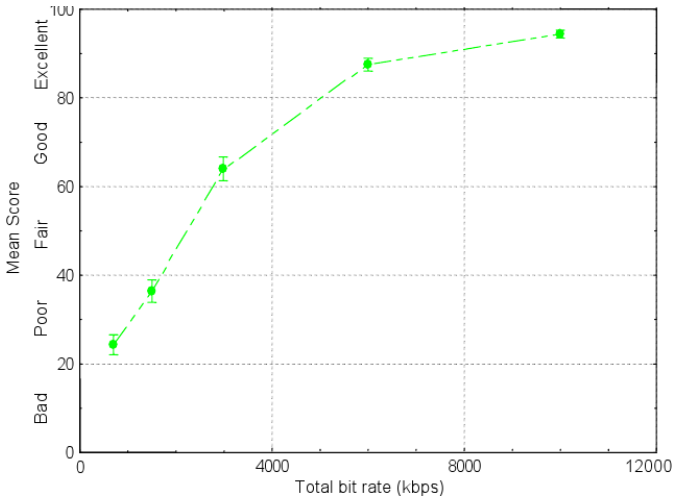

Figure 3. Recorded subjective quality assessment using the SAMVIQ scale as a function of encoding rate, averaged over all sequences. WM9 codec, SD video format.

is much steeper at lower rates than the CIF average within the corresponding region. A bit rate larger than 6 Mbps is not justified as the gain in terms of quality improvement is only minor.

C. Assessment of network parameters

The next task was to determine the network's influence on the perceived quality, i.e., the quality factor M_{N}. This was accomplished by exposing the compressed and streamed video signal to transmission errors. A network simulator emulated suitable conditions as typically encountered on real networks. More specifically, the effects of the error types delay, jitter, and packet loss were investigated.

A 60-second test sequence was assembled, consisting of six single short sequences of different characteristics, including the previously introduced TV material, which had been converted to CIF format (352×288 pixels) according to ITU-R BT.601 in advance. Hence, the results presented here are valid for a wide spectrum of source signals. The encoder and streaming server, namely the Microsoft Windows Media Encoder version 10 and Microsoft Windows Media Server 2003, were set up as explained in Section IV-B, using bit rate/frame rate combinations of 300 Kbps @12,5 fps, and {600, 1000, 1400} Kbps @25 fps. The key frame distance was set to 3 s, and the quality level to 50. The encoder established communication between client and server using the proprietary and now deprecated MMS (Microsoft Media Services) protocol version 9. With MMS, service requests are by default negotiated according to the Real-Time Streaming Protocol (RTSP), and the actual streaming relies on Real-Time Transport Protocol (RTP) and Real-Time Transport Control Protocol (RTCP), which are based on UDP.

The Shunra cloud network simulator (version 4.0) was used in the experiments. It exposed the bit stream to typical channel conditions and common error patterns, i.e. delay, jitter, and packet losses. While the maximum channel bandwidth was considered to be 1500 Kbps including return channel, the packet size was set to 1290 Bytes, both of which are realistic values. Prior to the experiment, the effect of delay, jitter, and packet loss were verified by means of the IxChariot network emulator and analyser, and the Ethereal (now Wireshark) analyser. The introduced error patterns were recorded by deploying the Shunra Cloud Catcher and Ethereal and fed as input into the network. By doing so, it was ensured that all bit streams were exposed to the same error pattern, providing equal network conditions with all encoder and streaming parameter settings.

As previously mentioned, all error scenarios included the error-free case, i.e., the transparent channel, and all error types were tested separately. Values for the delay of packets were {1200, 1400, 1800, 3000, 4000} ms, {400, 500, 600, 700} ms for jitter, and {1, 5, 10, 15, 20}% regarding channel packet losses. It should be stressed that these are packet losses on the network that are not necessarily identical to the packet losses encountered at the decoder, since reasonably large jitter and delay in combination with a small packet buffer at the decoder can lead to packet losses in addition to network packet losses. The buffer size is handled transparently by the player software.

For decoding, Windows Media Player version 10 with the default configuration was used. Decoding artifacts due to bit rate constraints and the effects of channel errors were typically blurring, blocking, and frozen image streams.

The subjective evaluation itself was conducted by six experts in the field of video processing in side-by-side comparisons with the encoded and decoded reference sequence (which can be treated as the case of an error-free transmission) on the one side, and the error-affected and decoded video on the other side. Among the shown sequences was also always a hidden reference sequence to check for any potential bias of the respective evaluator. The error types and error parameters were randomised in order. The evaluators ranked the perceived image quality according to the continuous MUSHRA-scale (Multi Stimulus test with Hidden Reference and Anchor), ranging from 0 to 100, with a splitting of the credit point scale into the five equally spaced categories as in the previous section. For each error setting, the evaluation's mean value and according 95% confidence interval were calculated.

Figure 4 summarises the findings for channel packet loss. The error-free case with 0% packet loss ratio shows how the quality of each stream is perceived depending only on the bit rate. Not surprisingly, the higher the bit rate, the better the quality. More specifically, the bars answer the question what bit rate is needed to achieve a "Fair" (40-60) or "Good" (60-80) quality with CIF imagery. The subjective quality is — as expected — identical for all error types in the error-free case.

As expected, the quality decreases with an increase of the packet loss ratio. It is interesting to see, however, that low-quality sequences are less affected by transmission errors and are hence more robust. In other words, the higher the image quality, the worse the degradation in case of errors. It appears that the best compromise between quality in the presence of network errors and quality in the error-free case is

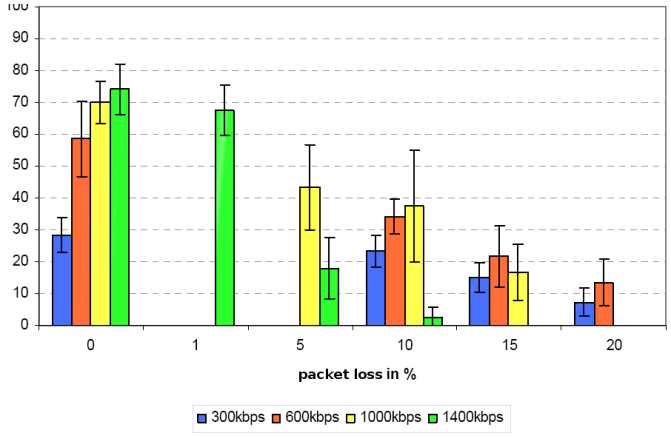

Figure 4. Quality assessment in SAMVIQ scale form as a function of bit rate and packet loss ratio. Not all combinations of rate and packet loss have been tested

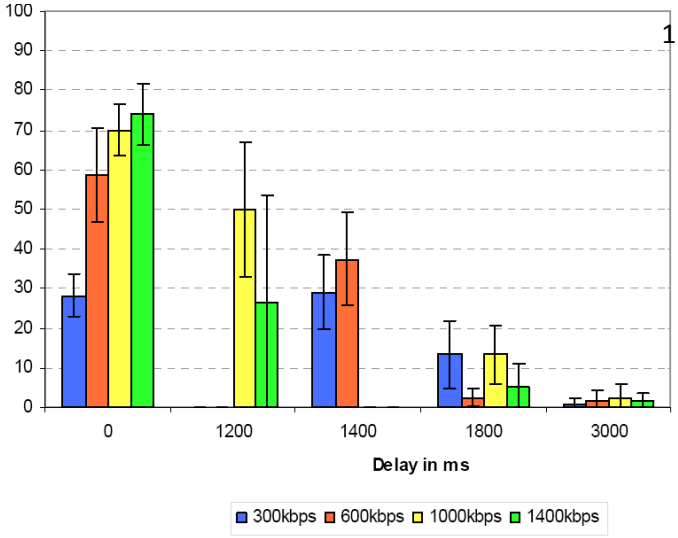

Figure 6. Quality assessment in SAMVIQ scale form as a function of bit rate and packet delay in ms. Not all combinations of rate and delay have been tested

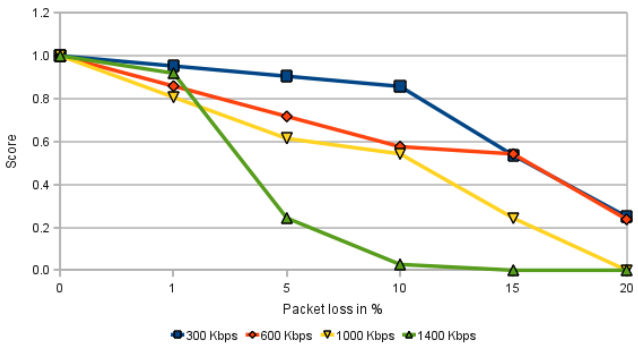

Figure 5. Quality factor M_{N}, depending on bit rate and packet loss

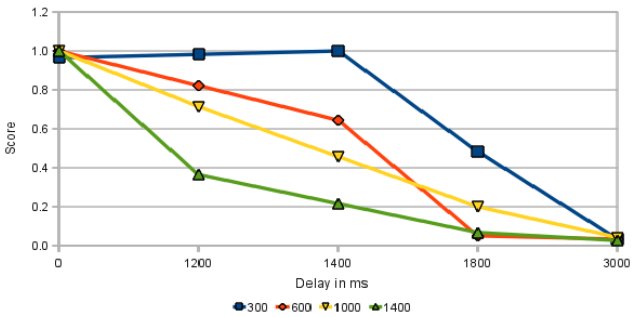

Figure 7. Quality factor M_{N}, depending on bit rate and packet delay

accomplished with a bit rate between 600 Kbps and 1000 Kbps with CIF-sized image material. Then, a packet loss ratio of roughly 5% is the critical threshold under which the channel conditions should not drop in order for the quality to remain "Fair".

From the perceived quality, the network quality factor M_{N} can easily be calculated. Given the maximum quality $Q_{\max}$ in the error-free case and the perceived quality $Q_{\mathrm{p}}(PLR)$ at a particular packet loss ratio (PLR), the factor can be defined as $Q_{\mathrm{p}}(PLR)/Q_{\max}$ and is hence mapped to the interval [1, 0]. The characteristics of the degradation of M_{N} with an increase of packet losses are shown in Figure 5. Missing measurement points have been linearly interpolated and extrapolated prior to the mapping from $Q_{\mathrm{p}}(PLR)$ to M_{N}.

We also show the results for jitter and packet delay for completeness, even though both error types will — depending on the size of the decoder's packet buffer — eventually lead to decoder packet losses. The characteristics of the curves are thus expected to be similar to the packet loss case, as also proven in Figure 6 and Figure 8.

Figure 6 shows the subjective quality in case of packet delay, while the normalised quality factor is plotted in Figure 7. As before, missing measurement values have been inter-/extrapolated before mapping $Q_{\mathrm{p}}(PLR)$ to M_{N}. The results are consistent with those in the packet loss scenario in that low-bitrate streams are inherently more robust to errors than streams with a high bit rate. With the aforementioned recommended bit rate in the range [600,1000] Kbps and the given CIF image sequences, the critical threshold, under which the channel conditions should not drop in order for the quality to remain "Fair", is approximately 1500 ms.

We have also performed experiments with a packet delay of 4000 ms and found results similar to the optimal performance, which is rather counter-intuitive, as we expected the quality to be lower than with a delay of 3000 ms. While our assessment setup is not designed to find the causes for such behaviour, possible explanations include: *(a)* The technical setup of the experiment might have a property that inflicts this behaviour, such as the network simulator shaping the traffic accordingly. Generating traffic patterns that are at the edge of the usual definition areas are likely to show unexpected behaviour and outliers. Such a pattern may interfere with the decoder's behaviour. An investigation of the traffic generator's behaviour and assessments with varying parameters other than the delay

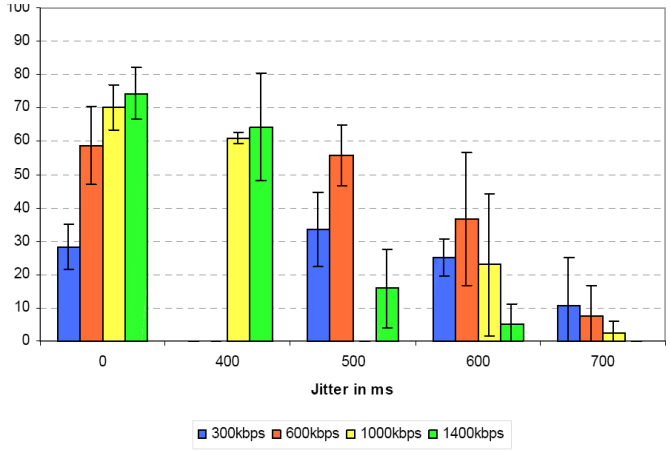

Figure 8. Quality assessment in SAMVIQ scale form as a function of bit rate and jitter in ms. Not all combinations of rate and jitter have been tested

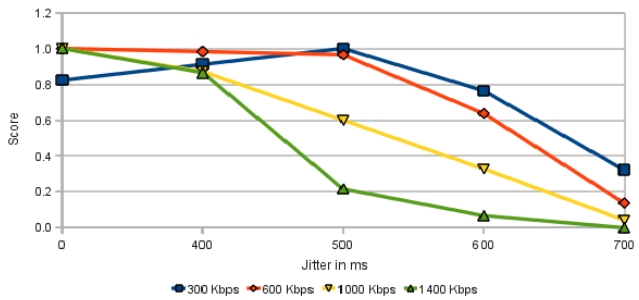

Figure 9. Quality factor M_{N}, depending on bit rate and jitter

value would be required to verify this assumption. *(b)* The video decoder software might have implemented functionality to avoid unfavourable network traffic patterns. Usually, some kind of buffering is used for this. Verifying this assumptions would require access to more precise technical data, or even the source code of the player software. *(c)* The decoder software also might have implemented active error recovery and concealment. A verification of this assumption would require either the complete technical specification of the player software or access to the source code. We have not been able to gain access to any of those due to the proprietory nature of the software. We exclude errors in the general setup of the experiment, since the SAMVIQ method was scientifically evaluated, and the experiment was performed under the supervision of the IRT who also were involved in the development of SAMVIQ.

Finally, Figure 8 summarises the findings for network jitter and neatly confirms the prior conclusion. Here, the critical threshold is roughly 600 ms. The normalised quality factor for jittering is plotted in Figure 9. The curve characteristics here are as explained above. The slight increase of M_{N} at a rate of 300 Kbps and a jitter of 500 ms as compared to the error-free case is most likely subject to the overall subjective impression of a number of image artifacts observed by the testing persons, such as blurring, blocking, and playback freezing, and should hence not be treated as being significant.

V. Controlling an Adaptive Streaming Application

In the following we apply the estimated subjective quality $\tilde{Q}$ to the Adaptive Internet Multimedia Streaming (ADIMUS) architecture. The ADIMUS architecture [36], [37] aims to support end-to-end video streaming of entertainment content with high media quality to mobile terminals. We are therefore concerned with data delivery mechanisms that impact quality. Since the content is streamed from live feeds or streaming servers to mobile devices, the interplay of backbone and access technologies is considered.

In our work, we consider a delivery system that consists of multiple service providers streaming video content via IP-based networks to multiple mobile terminals. These mobile terminals may use diverse network technologies and different types of terminals to access and view the content. As the video is transmitted from a service provider to a mobile terminal, its quality is degraded by several factors that are specific to different parts of the network infrastructure. Thus, we study the degradations originating from the network, and use the degradation values as input to the adaptation mechanism.

A. Adaptation in ADIMUS

The ADIMUS architecture comprises a delivery infrastructure based on source nodes at the service provider, an *overlay network* of ADIMUS proxies (AXs), and *multi-access networks* supporting mobile terminals as shown in Figure 10. The architecture contains QoS estimation mechanisms based on subjective (QoE) and objective (QoS) metrics from measured values.

1) The overlay network: In the backbone network, the data is routed through an *overlay network* which implements application-layer routing servers. To adapt to varying resource availability in the Internet, the AXs of the overlay network monitor connections and makes application-layer forwarding decisions to change routes. The overlay network of the streaming infrastructure uses appropriate streaming protocols and source-driven mechanisms for applying quality-improving mechanisms. Streaming servers representing source nodes, and the AXs operated by service providers, are placed in the Internet to form an overlay network. Such overlays constitute fully meshed networks that allow overlay re-routing when IP-based routing cannot maintain the required QoE on the direct IP route between server and client.

The AXs monitor network conditions using both passive and active network measurements, and they possibly interchange information about observed network conditions. Statistical information about the observed conditions of the network is used to estimate trends in, e.g., bandwidth, latency or packet loss at a given link or path. Note that routing decisions in the overlay network do not have the requirement of a very fast reaction time. This is by design because maintaining complete up-to-date bandwidth information does not scale. Instead, worst-case changes can increase the end-to-end delay

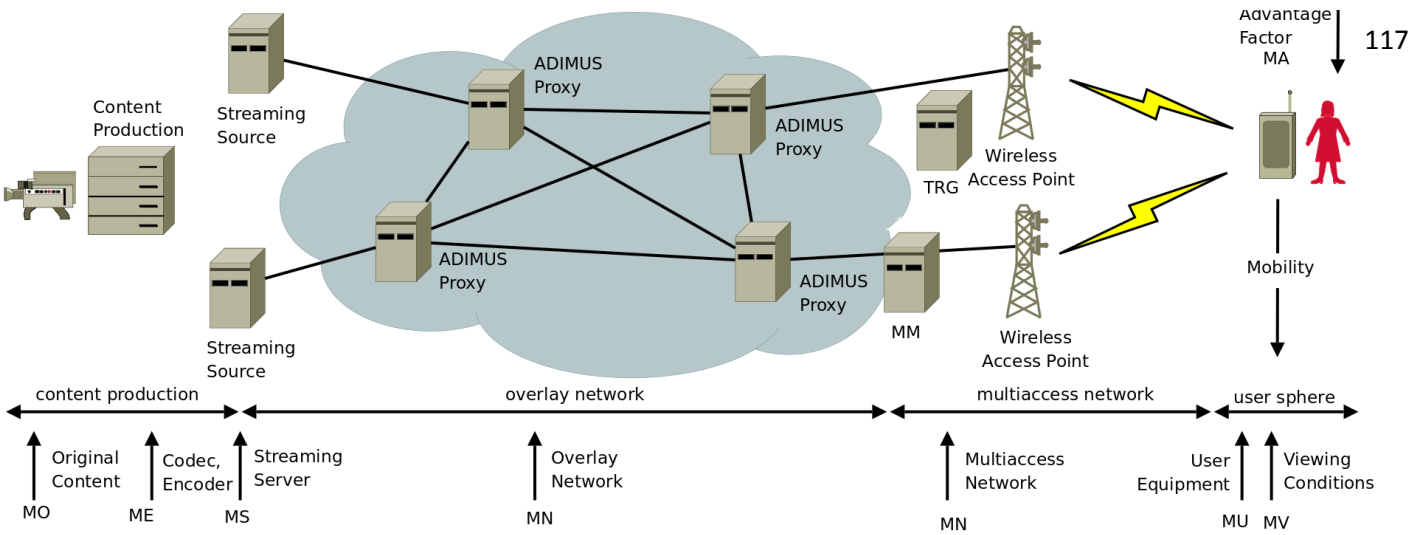

Figure 10. Transmission chain in the ADIMUS architecture. Entities which have an influence on the consumer's QoE are identified.

by a factor of several times. However, the full connectivity of the overlay network allows eventual recovery of the QoE if the required network resources are available, even if the QoE is temporarily not acceptable.

Terminals can initiate a session with the streaming server. Based on the overlay's monitor function of the network, the most appropriate overlay route is chosen. The last overlay node on the route always acts as a data source for the terminal. A re-invitation to the terminal is issued when route changes in the overlay involve a change in the last node. Such re-invitations are also used when route changes are triggered by reports from a mobile terminal.

Furthermore, ADIMUS supports multipath streaming to provide failure resistance and load balancing. Usually, the overlay nodes connected to the multi-access network use different access links for the different IP addresses of each of the mobile terminal's wireless devices. Aside from cross-layer information provided for faster reaction to changes at one of the wireless links, the multipath support is implemented entirely at the application level. This implies that multipath streaming is only possible to mobile terminals that run application-aware application layer software that handles buffering and reordering. The unavoidable reordering that occurs due to the use of several routes would lead to reduced QoE without buffering and re-ordering.

In each AX, algorithms perform the allocation of streams, so that the bandwidth is allocated, trying to optimise both the network load, and the subjective quality for the consumers. The algorithm in an AX can only make decisions on the basis of locally available information, and possibly on information that is communicated externally to an AX. In order to evaluate an algorithm for decision making at an AX, we can perform simulations for given topologies. The results of these simulations can then be compared to benchmarks that provide the optimal case [38], [39]. Since there are many possible scenarios we consider two major scenarios; the first being a unicast scenario where most user requests to the same video are not overlapped in time. This can be seen as a case for video-on-demand streaming. Multiple paths are built using the overlay nodes and optimal path selection and rate allocation is a challenge under varying network conditions. In a second scenario we consider multicast, which is suitable for life-streaming. The challenge here is to construct multiple multicast trees over the overlay network and perform the rate assignment to these trees.

2) The multi-access network: Near the mobile terminal a heterogeneous *multi-access network* provides application adaptation and handover mechanisms to maximise the Quality of Experience (QoE), and to support different types of mobility. A cross-layer signalling system is utilised to feed the different decision-points with continuous system status information. The multimedia streaming end point at the terminal-side is the mobile terminal, which resides in the multi-access network consisting of different IP-based access networks of different technologies and managed by different parties. The mobile terminal is equipped with multiple network interfaces, and has support for IP mobility protocols, such as Mobile IP (MIP) [40]. The mobile terminal is thus capable of roaming between IP networks. In the case of MIP, route optimisation needs to be supported for the mobile terminal to be able to update its mobility information directly to the correspondent node.

The multi-access network environment allows handovers as a means for maintaining the QoE. Specifically, the mobile terminal is capable of selecting an alternative access when the current link does not meet the minimum QoS requirements of our video streaming service. To make an informed handover decision, the mobile terminal collects and utilises information related to the available access options' characteristics, forming the degradation function $\tilde{Q}$. The parameters for $\tilde{Q}$ can be

obtained through a cross layer mechanism, such as the IEEE 802.21 Media Independent Handover (MIH) framework [41] and the triggering framework [42].

B. Applying the Degradation Function

We discuss how the estimated QoE $\tilde{Q}$ can be used to make decisions in ADIMUS, such as application layer routing decisions, transcoding, the selection of coded bandwidth, the selection of layers in the overlay network, and triggering decisions in the multi-access network.

In Section IV, we have shown that packet loss is the most relevant parameter to indicate the subjective quality. Therefore, when using $\tilde{Q}$ we make use of packet loss as the main parameter.

1) Packet Loss in the Overlay Network: The ADIMUS overlay network is based on the Internet infrastructure, and depends on the properties of Internet traffic in general. Since packet loss is also used to control other mechanisms on the Internet, such as the TCP rate control, we need to consider several scenarios to make sure that these control mechanisms on the Internet, and in ADIMUS, do not impact each other. In a network where TCP competes with other traffic, TCP rate control will increase bandwidth until packet loss happens; then it will reduce the bandwidth of a TCP stream. This mechanism will also affect packet loss of other streams, both UDP and TCP traffic. Since the strategy of TCP is to use as much bandwidth as possible, until packet loss occurs, TCP can affect other mechanisms in the Internet; it can also be affected by these. Random Early Detection (RED) [43] is one of these mechanisms that prohibit bandwidth saturation for TCP streams. On the other hand, other traffic, such as media streams, can impact the TCP rate control. Therefore, non-TCP traffic needs to be designed to be TCP-friendly [44].

In the Internet backbone, packet loss is usually a sign that the maximum bandwidth of a channel is exceeded, i.e., the Internet routers are not capable of forwarding all incoming packets. Different mechanisms in the Internet regulate which packets are forwarded and which packets are possibly discarded. Depending on the employed mechanisms the packet loss in the backbone has a certain burstiness, i.e., the ability to impact a larger series of packets. It is generally understood that packet loss on the Internet is likely to be bursty, but using multipath routing can render the loss patterns to be less bursty [45] if every packet sequence is distributed fairly to all paths.

When an AX is informed about too much packet loss in the overlay network (or in the backbone network) the reason is likely due to trying to use more capacity than is available. The overlay network can then try to re-route and thus avoid the links of the network that cause the packet loss. Alternatively, AXs in the network can throw away packets in an informed manner. When doing so the estimate of $\tilde{Q}$ for single consumers and A_{M} for groups of consumers can be used to optimise the possible user satisfaction under the given circumstances.

2) Forward Error Correction: ADIMUS is able to correct the media stream in each AX in order to achieve better QoE, at the cost of increased delay and jitter values. Forward Error Correction (FEC) [46] techniques can be applied to reduce the impact of packet loss. It is generally understood that the packet loss in the Internet is bursty, and using FEC is known to be ineffective to recover from burst errors. In this case, using multipath routing is beneficial as it can result in less bursty loss [45].

Using error concealment, missing packets can be replaced with corrected packets by the AX nodes in the application layer. Therefore, when using FEC, the packet loss rate for other traffic differs from the ADIMUS traffic. In practice, the estimated QoE in the application layer $\tilde{Q}_A$ is better than the estimated QoE in the network layer $\tilde{Q}_N$, i.e., $\tilde{Q}_A \geq \tilde{Q}_N$. However, we have shown in Section IV that loss degradation resilience for low-bandwidth streams is considerably higher than for high-bandwidth streams. The bandwidth of the original stream does therefore need to be taken into account in the decision to apply FEC.

An AX can limit ADIMUS stream bandwidth regardless whether the stream is FEC-protected. Considering the result graphs in Section IV, there is a trade-off, depending on the original encoding quality, represented by the coding bandwidth and the loss rate. Tolerating loss can, up to a certain threshold, lead to better subjective results than FEC-protecting the stream with lower coding bandwidth, provided that both streams consume the same amount of overlay resources. As an example, Figure 4 shows that the QoE decrease for lower coding bandwidth tends to be larger than the QoE decrease for tolerating higher loss rates.

3) Reserved Channels: Another solution to avoid interferences with other traffic is the use of reserved channels, e.g., mechanisms for DiffServ, IntServ, or MPLS tunnels [47]. In the case of reserved channels for ADIMUS, the value $\tilde{Q}$ can be used directly in the AXs to make routing decisions. However, the QoS mechanisms in the Internet need extra resources to be deployed by the ISPs. Therefore, these are currently not sufficiently deployed, while ISPs supporting these mechanisms will tend to have QoS reservations or reserved tunnels only as payable services. As a consequence, using these mechanisms could render ADIMUS only suitable for pay-services.

4) Packet Loss in the Multi-Access Network: In the multi-access network of ADIMUS packet loss is more likely caused by physical reasons in the wireless medium than being caused by exceeded bandwidth requirements. For instance, in WLAN networks the beacon can regularly cause a bursty packet loss, typically 60 consecutive packets every 1200 packets [32]. Therefore, in the multi-access network, packet loss cannot be considered an indicator for exceeded bandwidth limits. However, packet loss can still be used to estimate the quality of the stream using $\tilde{Q}$, and be included in the calculation of triggering decisions, i.e., switching between different base stations. A cross-layer signalling architecture related to ADIMUS for streaming to mobile terminals has been presented by Mäkelä et al. [48].

5) Estimated QoE for the Consumer: The transmission chain in ADIMUS, shown in Figure 10, can be considered to be similar in structure to the transmission chain shown in

Figure 1. Therefore, we adapt $\tilde{Q}$ to consider the influence of the overlay network, and the multi-access network in order to present the estimated QoE $\tilde{Q}$ for the consumer. For ADIMUS, $\tilde{Q}$ at the application layer needs to be considered, i.e., packet loss and other parameters that are measurable at the application layer.

$\tilde{Q}$ at the application layer gives an estimate on user satisfaction, and can show the need for adaptation in the network. However, $\tilde{Q}$ for the consumer does not give evidence on which adaptation measures need to be taken. These decisions are to be taken by cross-layer mechanisms in the multi-access network [48] and routing decisions in the overlay network. Thus, for ADIMUS we consider

$$M_{\mathrm{N}} = M_{\mathrm{N,overlay}} \cdot M_{\mathrm{N,multi\text{-}access}}$$

while the other settings are chosen similarly to the parameters shown in Section III.

C. Using Scalable Video Coding

In addition, to minimise the effects of transient congestion to multimedia transmission and QoE in a wireless access link, link-level adaptation can be used. In general, video streams consist of packets that have a differing impact on the decoded video quality. In the case of an MPEG-2 encoding, for example, packets contributing to I, P or B frames may be lost. Their absence affects the display quality for the playout duration of a whole group-of-pictures (GOP), part of a GOP, or only a single frame, respectively. Video streams can, in general, tolerate some packet loss, but this implies that loosing certain types of packets has a smaller impact on quality than loosing others. Rate-distortion analysis assesses this impact [49] and allows an AX to perform controlled dropping of the least important packets.

Scalable video codecs such as H.264 SVC [50] improve on the options that exist for controlled dropping. H.264 SVC encodes a base layer as an independent, and backward compatible stream that must be treated like a non-scalable stream. An SVC stream comprises additional enhancement layers that improve the temporal or spatial resolution of the base layer, or decrease its blurriness. These can be dropped by the AX without introducing any errors into the display of the base layer. However, the QoE can be reduced in a content-aware, controlled manner, since the AX can be instructed to avoid packet loss in the less important layers. Thus, it is possible to remove excess enhancement layers from the stream on the fly without affecting session continuity; the only effect is on the QoE. In the multi-access part of the ADIMUS architecture, link level adaptation is used to ensure that the terminal receives at least the most important frames, i.e., the base layer frames when under poor link conditions. In the overlay network of ADIMUS, using SVC the AX nodes can make content-aware decisions to drop packets that do not belong to the base layer.

The impact on the QoE when packet loss affects single layers of a scalable stream has not been studied here. However, further research is needed before SVC can be used for the algorithms employed in the AX nodes.

VI. Conclusion

We introduced an image and video quality degradation model to estimate the subjective image and video quality on the basis of objective measurements. The model takes all entities which potentially degrade the image and video quality into account. All influencing factors are either calculated for individual users or groups of consumers. The latter is achieved by adapting the APDEX method to the scenario.

The quality estimation model was implemented by means of pre-recorded tests for single consumers, the purpose of which is to represent realistic values in corresponding situations. The implementation uses look-up tables to calculate an estimate for the user's perceived image quality and can thereby provide help to locate the causes of reported quality degradation.

The results show how the perceived quality depends on a number of key parameters, among which the bit rate, content type, and packet loss ratio. Not surprisingly, the subjective quality decreases with an increase of the packet loss ratio. The degradation is much stronger for high-bandwidth than for low-bit-rate video, though. For packet loss, most of the degradation occurs with a packet loss ratio higher than 5%. This is deemed as rather large as compared to parameter values usually experienced in networks, but it is still an important result as such values may be encountered in real-life scenarios involving the users' own networking equipment. For the values of jitter and delay we experienced a similar result. Another important outcome is the image quality's dependency on the encoding and streaming bit rate. This allows the Internet service provider the possibility to adjust the bit rate automatically with regard to the observed packet loss ratio.

The degradation for different bit rates indicates clearly a more complex dependency between $M_{\mathrm{E,S}}$ and M_{N} than our simplified model allows. For the practical application of giving feedback to providers, we solved this problem by making the function for M_{N} dependent on discrete, fixed bandwidth settings. However, for the general case more research is needed. For a regression analysis, a much larger data sample, and thus assessment sessions, would be required.

The proposed model has been implemented and incorporated in current software that is in use at the Norwegian TV Channel TV2. In the implementation the values for packet loss, jitter, delay and system information are collected on the consumer's devices using an applet connected to the player software. This requires the consent of the consumer. These values are reported to a service installed at the CP or ISP, where the QoE values are calculated for each consumer and groups of consumers. The result of this calculation is shown graphically on a display at the CP or ISP. Since the measurement frequency for our application is in the range of seconds, the data rate of the return channel is low. The proposed model is of valuable help to content providers and Internet service providers to exclude causes of quality problems that are outside their control.

We discussed how to apply the estimated QoE and the degradation function to the ADIMUS architecture that sup-

ports streaming to mobile nodes. When controlling adaptation mechanisms in a streaming system, we need to identify possible interferences with other mechanisms in the Internet, such as the TCP rate control. Therefore, the estimated QoE cannot be used directly to control adaptation in a network or its parts. Instead, the estimated QoE can be used for decisions that an adaptation ought to take place.

Together with different application layer mechanisms, such as forward error correction or scalable video coding, the function $\tilde{Q}$ can give valuable hints for decision making in the adaptation mechanisms, especially for the cases where nodes, here the AXs, need to make decisions on the basis of local knowledge. Since the AX nodes can do error concealment at the application layer, $\tilde{Q}$ for an AX can help make decisions for optimising the overlay network.

Algorithms that make routing decisions in the nodes of the overlay network need to use $\tilde{Q}$ as input. However, research is needed which weight factors are useful for different topologies and other settings in the ADIMUS architecture. The careful design of these algorithms and the necessary parameters will be done, and compared to benchmarks that already have been developed.

ACKNOWLEDGMENT

The work described in this document has been conducted as part of the MOVIS project funded by the Norwegian Research Council, and the ADIMUS (Adaptive Internet Multimedia Streaming) project funded by the NORDUnet-3 programme. The authors wish to thank Volker Steinmann and in memoriam Gerhard Stoll at the IRT for performing the assessment for the MOVIS project. We want to thank Arne Berven at the Norwegian TV channel TV2, and Reza Shamshirgaran at Nimsoft AS for the discussions during the implementation of the feedback system developed in this project. Finally, we thank Tiia Sutinen, Knut Holmqvist, Trenton Schulz and Carsten Griwodz for discussions during the course of the ADIMUS project and the preparation of this paper.

REFERENCES

[1] W. Leister, S. Boudko, and T. Halbach, "Estimation of subjective video quality as feedback to content providers," *Proc. ICSNC'10, Fifth International Conference on Systems and Networks Communication*, pp. 266–271, 2010.

[2] Norwegian Computing Center, "MOVIS — Perfomance monitoring system for video streaming networks," Web page, Sep. 2007, accessed 2010-11-29. [Online]. Available: http://www.nr.no/pages/dart/project_flyer_movis

[3] Tektronix, "A guide to maintaining video quality of service for digital television programs," White Paper, Tektronix Inc, Tech. Rep., 2000.

[4] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis, "Framework for IP Performance Metrics," RFC 2330, May 1998, [Online]. Available: http://www.ietf.org/rfc/rfc2330.txt, last accessed May 14, 2010.

[5] V. Smotlacha, "QoS oriented measurement in IP networks," CESNET Report 17/2001, CESNET, Report 17/2001, 2001, [Online]. Available: http://www.cesnet.cz/doc/techzpravy/2001/17/qosmeasure.pdf, last accessed May 14, 2010.

[6] D. P. Pezaros, "Network traffic measurement for the next generation Internet," Ph.D. dissertation, Computing Dept. Lancaster University, 2005.

[7] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electronic Letters*, vol. 44, no. 23, pp. 800–801, 2008.

[8] P. Ni, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Fine-grained scalable streaming from coarse-grained videos," in *International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV)*. ACM, 2009, pp. 103–108.

[9] M. Knee, "The picture appraisal rating (PAR) - a single-ended picture quality measure for MPEG-2," Snell & Wilcox Limited, White Paper, 2006, [Online]. Available: http://www.snellgroup.com/documents/white-papers/white-paper-picture-appraisal-rating.pdf, last accessed May 14, 2010.

[10] H. Koumaras, A. Kourtis, C.-H. Lin, and C.-K. Shieh, "End-to-end prediction model of video quality and decodable frame rate for MPEG broadcasting services," *International Journal On Advances in Networks and Services*, vol. 1, no. 1, pp. 19–29, 2009, [Online]. http://www.iariajournals.org/networks_and_services/ last accessed March 25, 2010.

[11] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 246–250, 1997.

[12] S. Wolf and M. Pinson, "Video quality measurement techniques," NTIA Report 02-392, National Telecommunications and Information Administration, US Dept of Commerce, NTIA Report 02-392, 2002.

[13] M. Mu, "An interview with video quality experts," *SIGMultimedia Rec.*, vol. 1, no. 4, pp. 4–13, 2009.

[14] M. H. Pinson and S. Wolf, "Comparing subjective video quality testing methodologies," in *VCIP*, ser. Proceedings of SPIE, T. Ebrahimi and T. Sikora, Eds., vol. 5150. SPIE, 2003, pp. 573–582.

[15] ——, "An objective method for combining multiple subjective data sets," in *VCIP*, ser. Proceedings of SPIE, T. Ebrahimi and T. Sikora, Eds., vol. 5150. SPIE, 2003, pp. 583–592.

[16] ITU-R, "Methodology for the subjective assessment of the quality of television pictures," ITU Standardization Document, ITU, Genève, ITU Standardization Document, 2004.

[17] EBU project group B/VIM, "SAMVIQ - subjective assessment methodology for video quality," European Broadcasting Untion (EBU), Report BPN 056 / B/VIM 053, 2003.

[18] F. Kozamernik, P. Sunna, E. Wyckens, and D. I. Pettersen, "Subjective quality of internet video codecs," European Bradcasting Union (EBU), EBU Technical Review, 2005.

[19] F. Kozamernik, V. Steinmann, P. Sunna, and E. Wyckens, "QSAMVIQ - a new EBU methodology for video quality evaluations in multimedia," *Proc. IBC 2004*, 2004.

[20] J. Bennett and A. Bock, "In-depth review of advanced coding technologies for low bit rate broadcast applications," in *proc. IBC 2003*. IBC, 2003, Proc. IBC 2003, pp. 464–472.

[21] P. Casagranda and P. Sunna, "Migration of new multimedia compression algorithms to broadband applications," in *proc. IBC 2003*, 2003, proc. IBC 2003, pp. 473–479.

[22] P. Ni, A. Eichhorn, C. Griwodz, and P. Halvorsen, "Frequent layer switching for perceived quality improvements of coarse-grained scalable video," *Multimedia Syst.*, vol. 16, no. 3, pp. 171–182, 2010.

[23] N. Cranley, P. Perry, and L. Murphy, "User perception of adapting video quality," *International Journal of Man-Machine Studies*, vol. 64, no. 8, pp. 637–647, 2006.

[24] NetPredict Inc., *Assess the Ability of Your Network to Handle VoIP before You Commit*, White Paper, NetPredict Inc, 2002-2004.

[25] ITU Study Group 12, *Perceptual evaluation of speech quality (PESQ) – An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.*, ITU-T Recommendation P.862, ITU-T, 2001.

[26] C. Hoene, "Internet telephony over wireless links," TU Berlin, Tech. Rep., 2005.

[27] ITU-T, *Single Ended Method for Objective Speech Quality Assessment in Narrow-Band Telephony Applications*, ITU-T Recommendation P.563, ITU-T, 2005.

[28] ——, *G.107: The E-model, a computational model for use in transmission planning*, ITU-T Recommendation, ITU-T, 2003.

[29] "Apdex - Application Performance Index," web pages www.apdex.org, last accessed March 25, 2010.

[30] P. Sevcik, "Defining the application performance index," *Business Communications Review*, pp. 8–10, March 2005.

[31] S. Katsuno, T. Kubo, K. Yamazaki, and H. Esaki, "Measurement and analysis of multimedia application and IPv6 ADSL Internet access network," *Proc. of the 2003 Symposium on Applications and the Internet (SAINT'03)*, 2003.

[32] D. Pezaros, D. Hutchison, F. Garcia, R. Gardner, and J. Sventek, "Service quality measurements for IPv6 inter-networks," *Proceedings of the 12th International Workshop on Quality of Service IEEE*, 2004.

[33] B. Sat and B. W. Wah, "Playout scheduling and loss-concealments in voip for optimizing conversational voice communication quality," in *ACM International Multimedia Conference (ACM MM)*, Oct. 2007, pp. 137–146.

[34] V. Steinmann, A. Vogl, and G. Stoll, *Subjective assessment of video quality depending on encoding parameters, Contribution of IRT to MOVIS WP2 Phase 1*, technical documentation, Institut für Rundfunktechnik, 2006.

[35] G. Stoll, A. Vogl, and V. Steinmann, *Subjective Assessment of Video Quality depending on networking parameters*, technical documentation, Institut für Rundfunktechnik, 2007.

[36] W. Leister, T. Sutinen, S. Boudko, I. Marsh, C. Griwodz, and P. Halvorsen, "An architecture for adaptive multimedia streaming to mobile nodes," in *MoMM '08: Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*. New York, NY, USA: ACM, 2008, pp. 313–316.

[37] ——, "ADIMUS – Adaptive Internet Multimedia Streaming – final project report," Norsk Regnesentral, Oslo, Report 1026, September 2010, [Online]. Available: http://publ.nr.no/5335, last accessed January 11, 2011.

[38] S. Boudko, C. Griwodz, P. Halvorsen, and W. Leister, "A benchmarking system for multipath overlay multimedia streaming," in *Proceedings of ICME*, 2008.

[39] ——, "Maximizing video quality for several unicast streams in a multipath overlay network," in *Proc. International Conference on Internet Multimedia Systems Architecture and Application 2010, Bangalore, India*, A. Dutta and S. Paul, Eds. IEEE, 2010.

[40] C. Perkins, Ed., *Mobile IP: Design Principles and Practice*. Addison-Wesley, 1998.

[41] IEEE, "Media Independent Handover," IEEE Draft Standard 802.21, 2008, work in progress.

[42] J. Mäkelä and K. Pentikousis, "Trigger management mechanisms," in *Proceedings of ISWPC*, San Juan, Puerto Rico, February 2007, pp. 378–383.

[43] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 397–413, 1993.

[44] J. Widmer, R. Denda, and M. Mauve, "A survey on TCP-friendly congestion control," *IEEE Network*, vol. 15, pp. 28–37, 2001.

[45] T. Nguyen, P. Mehra, and A. Zakhor, "Path diversity and bandwidth allocation for multimedia streaming," in *proceedings of ICME*, 2003, pp. 6–9.

[46] Y. Wang and Q. fan Zhu, "Error control and concealment for video communication: A review," in *Proceedings of the IEEE*, 1998, pp. 974–997.

[47] V. Sharma and F. Hellstrand, "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery," RFC 3469 (Informational), Feb. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3469.txt

[48] J. Mäkelä, M. Luoto, T. Sutinen, and K. Pentikousis, "Distributed information service architecture for overlapping multiaccess networks," *Multimedia Tools and Applications*, pp. 1–18, 2010, 10.1007/s11042-010-0589-9.

[49] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, no. 6, pp. 74–90, Nov. 1998.

[50] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable H.264/MPEG4-AVC extension," in *Proceedings of ICIP*, October 2006, pp. 161–164.

[51] T. Ebrahimi and T. Sikora, Eds., *Visual Communications and Image Processing 2003*, ser. Proceedings of SPIE, vol. 5150. SPIE, 2003.

A Runtime Testability Metric for Dynamic High-Availability Component-based Systems

Alberto Gonzalez-Sanchez, Éric Piel, Hans-Gerhard Gross, and Arjan J.C. van Gemund
Department of Software Technology
Delft University of Technology
Mekelweg 4, 2628CD Delft, The Netherlands
{*a.gonzalezsanchez, e.a.b.piel, h.g.gross, a.j.c.vangemund*}*@tudelft.nl*

***Abstract*—Runtime testing is emerging as the solution for the integration and assessment of highly dynamic, high availability software systems where traditional development-time integration testing cannot be performed. A prerequisite for runtime testing is the knowledge about to which extent the system can be tested safely while it is operational, i.e., the system's *runtime testability*. This article evaluates Runtime Testability Metric (RTM), a cost-based metric for estimating runtime testability. It is used to assist system engineers in directing the implementation of remedial measures, by providing an action plan which considers the trade-off between testability and cost. We perform a theoretical and empirical validation of RTM, showing that RTM is indeed a valid, and reasonably accurate measurement with ratio scale. Two testability case studies are performed on two different component-based systems, assessing RTM's ability to identify runtime testability problems.**

***Keywords*-Runtime testability, runtime testing, measurement, component-based system.**

I. Introduction

Integration and system-level testing of complex, high-available systems is becoming increasingly difficult and costly in a development-time testing environment because system duplication for testing is not trivial. Such systems have high availability requirements, and they cannot be put off-line to perform maintenance operations, e.g., air traffic control systems, emergency unit systems, banking applications. Other such systems are dynamic Systems-of-Systems, or service-oriented systems for which the sub-components are not even known a priori [2], [3].

Runtime testing [4] is an emerging solution for the validation and acceptance testing for such dynamic high-availability systems, and a prerequisite is the knowledge about which items can be tested safely while the system is operational. This knowledge can be expressed through the concept of *runtime testability* of a system, and it can be referred to as *the relative ease and expense of revealing software faults*.

Figure 1 depicts this fundamental difference between traditional integration testing and runtime testing. On the left-hand side, a traditional off-line testing method is used, where a copy of the system is created, the reconfiguration is planned, tested separately, and once the testing has finished the changes are applied to the production system. On the right-hand side, a runtime testing process where the planning and testing phases are executed over the production system.

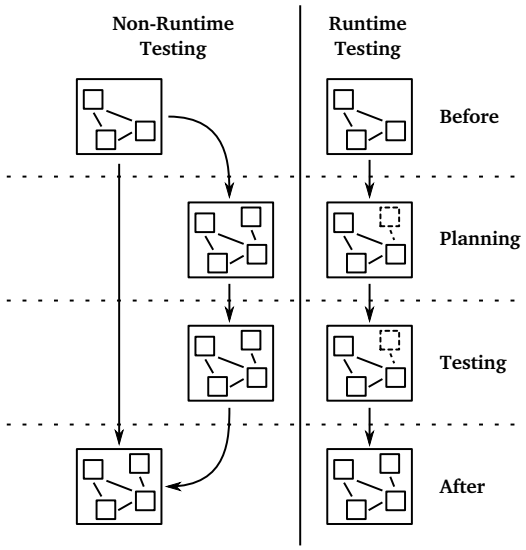

Figure 1. Non-runtime vs. runtime testing

Testability enhancement techniques have been proposed either to make a system less prone to hiding faults [5], [6], [7], or to select the test cases that are more likely to uncover faults with the lowest cost [8], [9], [10], [11]. However, they are not suited for the specific challenges posed by runtime testing, especially the cost that the impact tests will cause on the running system, which determines the viability of runtime testing, is not taken into account by those methods. Features of the system that need tests and whose impact cost is too high will have to be left untested, increasing the probability of leaving uncovered faults. Knowledge of the impact that runtime tests will have on the system will allow engineers to select and implement the appropriate needed measures to avoid interference with the system, or with its environment. As more features can be runtime tested, the probability of uncovering integration faults in the system increases.

This paper evaluates the Runtime Testability Metric (RTM) introduced in our earlier work [12], [13]. The metric reflects the trade-off that engineers have to consider, between the improvement of the runtime testability of the system after some interferences are addressed, and the cost of the remedial measures that have to be applied. The two main contributions of the paper are (1) the measurement-theoretical characterisation of RTM and its empirical validation, and (2) and evaluation of this metric on two industrial systems. In addition, scalable algorithm is introduced to calculate the near-optimal *action plan*, which list by effectiveness the operations that must become testable to improve the RTM.

The paper is structured as follows. In Section II runtime testability is defined. In Section III its theoretical characterisation is performed. An empirical validation of the metric is presented in Section IV. Section V evaluates the RTM on two example cases. In Section VI, an approximate, scalable algorithm is presented for the calculation of the action plan. Section VII describes an implementation of our metric into a component framework. Finally, Section IX presents our conclusions and future plans.

II. Runtime Testability

RTM addresses the question to which extent a system may be runtime tested without affecting it or its environment. Following the IEEE definition of testability [14], runtime testability can be defined as (1) the extent to which a system or a component facilitates runtime testing without being extensively affected; (2) the specification of which tests are allowed to be performed during runtime without extensively affecting the running system. This considers (1) the characteristics of the system and the extra infrastructure needed for runtime testing, and (2) the identification of which test cases are admissible out of all the possible ones. An appropriate measurement for (1) provides general information on the system independent of the nature of the runtime tests that may be performed, as it is proposed in [6], [7] for traditional testing. A measurement for (2) will provide information about the test cases that are going to be performed, as proposed in [8], [9], [10]. Here, we concentrate on (1), in the future, we will also consider (2).

Runtime testability is influenced by two characteristics of the system: *test sensitivity*, and *test isolation* [15]. Test sensitivity characterises features of the system suffering from test interference, e.g., existence of an internal state in a component, a component's internal/external interactions, resource limitations. Test isolation is applied by engineers in order to counter the test sensitivity, e.g., state duplication or component cloning, usage of simulators, resource monitoring. Our approach consists in performing an analysis of which features of the system present test sensitivity, prior to the application of isolation measures.

Moreover, testability can also be affected by the design and code quality, which can be measured in terms of robustness, maintainability, flexibility... Here we do not take into account these additional factors and concentrate on the behavioural, and most prevalent, factor for runtime testing: test interference.

The generic aspect of RTM allows engineers to tailor it to their specific needs, applying it to any abstraction of the system for which a coverage criterion can be defined. For example, at a high granularity level, coverage of function points (as defined in the system's functional requirements) can be used. At a lower granularity level, coverage of the component's state machines can be used, for example for *all-states* or *all-transitions* coverage.

In the following, we will precisely define RTM in the context of component-based systems.

A. Model of the System

Component-based systems are formed by components bound together by their service interfaces, which can be either provided (the component offers the service), or required (the component needs other components to provide the service). During a test, any service of a component can be invoked, and the impact that test invocation will have on the running system or its environment is represented as cost. This cost can come from multiple sources (computational cost, time or money, among others).

Operations whose impact (cost) is prohibitive, are designated as untestable. This means that a substantial additional investment has to be made to render that particular operation in the component runtime testable.

In this paper we will abstract from the process of identifying the cost sources, and we will assume that all operations have already been flagged as testable or untestable. In reality, this information is derived from an analysis of the system design and it environment. This latter analysis is performed by the system engineers, who have the proper domain-specific knowledge. Future research will address the issue of deriving this cost information, and of deciding whether a certain impact cost is acceptable or not.

In order to apply RTM, the system is modelled through a Component Interaction Graph (CIG) [16]. A CIG is defined as a directed graph with weighted vertices, $CIG = \langle V, V_0, E, c \rangle$, where

- $V \equiv V_P \cup V_R$: vertices in the graph, formed by the union of the sets of provided and required operations by the components' interfaces.
- $V_0 \subseteq V$: input operations to the system, i.e., operations directly accessible to test scripts.
- $E \subseteq V \times V$: edges in the graph, representing dependencies between operations in the system. E.g., if $(v_1, v_2) \in E$, v_1 depends on v_2.
- $c : V \to \mathbb{R}^+$: function that maps a specific operation to the preparation cost that is going to be optimised.

Each vertex $v_i \in V$ is annotated with a testability flag τ_i, meaning whether the cost of traversing such vertex (i.e.,

invoking that service) when performing runtime testing is prohibitive or not, as follows:

$$\tau_i = \begin{cases} 1 & \text{if the vertex can be traversed} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Edge information from within a component can be obtained either by static analysis of the source code, or by providing state or sequence models [16]. Inter-component edges can be derived from the runtime connections between the components. In the case no information is available for a certain vertex, a conservative approach should be taken, assigning $\tau_i = 0$.

Example CIGs are shown in Fig. 6 and Fig. 7. Nodes or vertices of the CIG represent component operations annotated with a testability flag, i.e., small black testable, large red crossed untestable. Edges of the CIG represent (1) provided services of a component that depend on required services of that same component (intra-component); and (2) required services of a component bound to the actual provider of that service (inter-component).

B. Estimation of RTM

RTM is estimated in terms of impact cost of covering the features represented in the graph. We do not look at the concrete penalty of actual test cases, but at the possible cost of a test case trying to cover each element. Because CIG is a static model, assumptions have to be made on the actual behaviour of test cases. In the future, we will enrich the model with additional dynamic information to relax these assumptions. Because of lacking control flow information, there is no knowledge about edges in the CIG that will be traversed by a test case. In the worst case, the interaction might propagate through all edges, affecting all reachable vertices. For the moment, we assume this worst case: assume that all the vertices reachable from v_i, which we will denote as P_{v_i} (predecessors set), can be affected.

Following our assumptions, the total preparation cost needed to involve an operation in a runtime test, taking into account the individual preparation costs of all the operations it depends on, is defined as

$$C(v_i) = \sum_{v_j \in S_{v_i}} c(v_j) \quad (2)$$

where v_i and v_j are operations, and S_{v_i} is the set of successors of vertex v_i, i.e., all the vertices reachable from v_i including v_i itself.

Not all operations can be directly tested, only a subset of possible input vertices V_0 can be reached directly. Other operations are reached indirectly, via a sequence of operations, which necessarily starts with an operation in V_0. We model this by only counting operations that can be reached from a testable input vertex $v_0 \in V_0$, i.e., that $v_i \in S_{v_0}$ and whose $C(v_0) = 0$. RTM can be then defined as

$$\begin{aligned} RTM = & |\{v_i \in V : C(v_i) = 0 \wedge \\ & \exists v_0 \in V_0 : v_i \in S_{v_0} \wedge C(v_0) = 0\}| \end{aligned} \quad (3)$$

This value can be divided by $|V|$ in order to obtain a normalised metric $rRTM$ that one can use to compare the runtime testabilities of systems with different number of vertices. However, such application has important implications on the theoretical requirements on the metric, as we will observe in Section III.

C. Improving the System's RTM

Systems with a high number of runtime untestable features (i.e., low runtime testability) can be improved by applying isolation techniques to specific vertices, to bring their impact cost down to an acceptable level. However, not all interventions have the same cost, nor do they provide the same gain. Ideally, the system tester would plot the improvement of runtime testability versus the cost of the fixes applied, in order to get full information on the trade-off between the improvement of the system's runtime testability and the cost of such improvement. This cost depends on the isolation technique employed: adaptation cost of a component, development cost of a simulator, cost of shutting down a part of the system, addition of new hardware, etc. Some of those costs will be very small because they correspond to trivial fixes. However, there can be extremely high costs that will make providing a fix for that specific component prohibitive. For example, a test of an update of the software of a ship that can only be performed at the shipyard has a huge cost, because the ship has to completely abandon its normal mission to return to dry dock. Even though these costs involve diverse magnitudes (namely time and money), for this paper we will assume that they can be reduced to a single numeric value: c_i.

III. Theoretical Validation

In this section, we establish the characteristics of the RTM measurement from a measurement-theoretical point of view. It allows us to identify what statements and mathematical operations involving the metric and the systems it measures are meaningful and consistent. We will concentrate (1) on RTM's fundamental properties, and (2) on its type of scale.

A. Fundamental Properties

In this section, we will study the properties required for any measurement. These properties determine whether RTM fulfils the minimal requirements of any measurement, i.e., whether it actually creates a mapping between the desired empirical property and the characteristics of the system, that can be used to classify and compare systems. The properties were described by Shepperd and Ince in [17] through an axiomatic approach.

Axiom 1: It must be possible to describe the rules governing the measurement.

This is satisfied by the formal definition of RTM and the CIG.

Axiom 2: The measure must generate at least two equivalence classes.

$$\exists p, q \in CIG : RTM(p) \neq RTM(q)$$

In Figure 2 are assigned two different RTM values, therefore proving this axiom. A node marked with a $\times$ represents an operation where $c(v) > 0$.

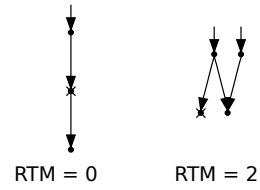

Figure 2. Two equivalence classes for RTM

Axiom 3: An equality relation is required.

This axiom is satisfied given that our measurement is based on natural numbers, for which an equality relation is defined.

Axiom 4: There must exist two or more structures that will be assigned the same equivalence class.

$$\exists p, q \in CIG : RTM(p) = RTM(q)$$

Figure 3 shows a collection of systems belonging to the same equivalence class, proving this axiom.

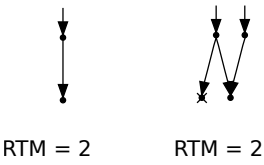

Figure 3. Equivalent CIGs for RTM

Axiom 5: The metric must preserve the order created by the empirical property it intends to measure. This axiom is also known as the Representation Theorem.

$$\forall p, q \in CIG : p \underset{rt}{\succeq} q \Leftrightarrow RTM(p) \geq RTM(q)$$

where $\underset{rt}{\succeq}$ represents the empirical relation 'more runtime testable than'.

This last axiom means that for any two systems, the ordering produced by the empirical property "runtime testability" has to be preserved by RTM. It is possible to find systems for which this axiom does not hold for RTM, because of the assumptions that had to be made (see Section II-A). However, we can empirically assess the effect of these assumptions on the consistency and accuracy of RTM. An empirical study about the accuracy of RTM is presented in Section IV.

B. Type of Scale

The theoretical characterisation of the metric's scale type (i.e., ordinal, interval, ratio, absolute) determines which mathematical and statistical operations are meaningful. This is important, because certain optimisation algorithms require specific mathematical operations that might not be meaningful for RTM, for example for defining heuristics as we will see in Section VI.

Assuming RTM satisfies Axiom 5, i.e., it preserves the empirical ordering of runtime testability, then, by definition, RTM defines a homomorphism from runtime testability to the Natural numbers. Therefore, by the ordered nature of Natural numbers, we can assert that RTM can be used as an *ordinal scale* of measurement. In practice this is true only for systems in which our assumptions about control flow and dependencies hold.

In order to be able to use RTM as a ratio scale measurement, in addition to the requirements for the ordinal type of scale being satisfied, a concatenation operation with an additive combination rule [18] must exist. A meaningful concatenation operation is creating the union of both systems by disjoint union of their CIG models. This operation, $\cup : CIG \times CIG \rightarrow CIG$, can be defined as $A \cup B = \langle V, V_0, E, c \rangle$, where

- $V \equiv V_A \cup V_B$
- $V_0 \equiv V_{0A} \cup V_{0B}$
- $E \equiv E_A \cup E_B$
- $c(v) = \begin{cases} c_A(v) & \text{if } v \in V_A \\ c_B(v) & \text{if } v \in V_B \end{cases}$

For this concatenation rule, the additive combination rule

$$RTM(A \cup B) = RTM(A) + RTM(B) \qquad (4)$$

can be used. Therefore, RTM can be used as a ratio scale with extensive structure (e.g., like mass or length), with respect to the disjoint union operation.

C. Relative Values

If we divide the values of RTM by the total number of operations, $|V|$, we can obtain the relative runtime testability ($rRTM$), to compare systems in relative terms. This transforms the measurement from a count into a ratio. Ratios, as percentages have an absolute scale [18] and cannot be combined additively.

For the runtime testability ratio and disjoint union concatenation operator, we can define the combination rule

$$rRTM(A \cup B) = \alpha \cdot rRTM(A) + (1 - \alpha) \cdot rRTM(B) \quad (5)$$

where $\alpha = \frac{|V_A|}{|V_A| + |V_B|}$.

This combination rule is not additive, in order to state the effect of a combination of two systems we need more information than RTM, namely the size relation α.

D. Summary and Implications

Because we proved that RTM fulfils the minimal properties of any measurement, RTM can be used to *discriminate* and *equalise* systems. Therefore, the statements '*system A has a different runtime testability than B*', and '*systems A and B have the same runtime testability*', are meaningful. Moreover, as we proved RTM has an ordinal scale type, RTM can be used to *rank* systems. The statement '*system A has more runtime testable operations than B*' becomes meaningful, and this enables us to calculate the median of a sample of systems, and Spearman's rank correlation coefficient.

Furthermore, by proving the ratio scale for RTM, it can also be used to *rate* systems, making the statement '*system A has X times more runtime testable operations than B*' a meaningful one. This allows performing a broad range of statistic operations meaningfully, including mean, variance, and Pearson's correlation coefficient.

RTM can also be used alone to reason about the composition of two systems. Due to its additive combination rule, '*systems A and B composed, will be* $RTM(A) + RTM(B)$ *runtime testable*' is a meaningful statement, provided that A and B are disjoint. This is not true for the relative $rRTM$, as additional information about the relationship between the systems (α, the size relation between the systems) is needed.

In order to support the theory, we are presenting an empirical validation with comparison with other metrics.

IV. Empirical Validation

In this section we conduct a number of experiments in order to empirically determine how accurate RTM is with respect to the empirical property of "runtime testability" (ERT).

A. Experimental Setup

To obtain the value of RTM, vertices are first classified into testable and untestable by means of $C(v)$ (see Eq. 2). Our goal is to assess the influence of the assumptions made when defining RTM, in the number of false positives and false negatives of this classification, and in the final value of RTM.

In order to have a baseline for comparison, the naive approach of just counting directly testable operations was used, defined as:

$$NTES = |\{v_i \in V : c(v_i) = 0\}| \quad (6)$$

We also use a previous proposition of RTM [15], which we name RTM_{old} and is defined as:

$$RTM_{old} = |\{v_i \in V : C(v_i) = 0\}| \quad (7)$$

Two systems were used in the experiment: AISPlot and WifiLounge, which are detailed in Section V. For the experiment, 500 variations of each system with different RTM values were generated by choosing the untestable vertices by randomly sampling in groups of increasing size from 2 to 30 untestable vertices.

The value of ERT to perform the comparison was obtained by creating and executing an exhaustive test suite in terms of vertices and execution paths. The set of operations covered when executing each test case was recorded. If a test case used any untestable operation, none of the operations covered by the test were counted. The test cases covered both systems completely, and redundantly, by exercising every possible path in the CIG from every input operation of the system. This way it was ensured that if an operation was not covered, it was not because a test case was missing, but because there was no test case that could cover it without requiring also an untestable operation.

B. Results

From each system and metric pair in this experiment, we recorded the following data:

- M_{set}: Set of operations classified as testable.
- Cov: Set of operations covered.
- $f_p = (|M_{set} - Cov|)/|M_{set}|$: false positive rate, i.e., operations wrongly classified as testable.
- $f_n = (|Cov - M_{set}|)/|Cov|$: false negative rate, i.e., operations wrongly classified as untestable.
- $\bar{e} = ||M_{set}| - |Cov||$: absolute error between the predicted and empirical runtime testabilities.

	System	f_p	f_n	$\bar{e}$
NTES	AISPlot	0.942	0.000	83.487
	WifiLounge	0.713	0.000	57.093
RTM_{old}	AISPlot	0.882	0.107	15.012
	WifiLounge	0.577	0.079	27.664
RTM	AISPlot	0.411	0.111	2.418
	WifiLounge	0.306	0.128	9.101

Table I
FALSE POSITIVE/NEGATIVE RATE, AND ERROR

Table I shows the rates of false positives and false negatives, along with the absolute error, averaged over 500 runs. The deviation between the predicted testability and the actual covered operations for each sample can be seen in the three plots in Figure 4. The dashed line represents the ideal target. Any point above it, constitutes an overestimation error, and below it, an underestimation error.

NTES: It can be seen that NTES has an extremely high error caused by its high false positive rate (94% and 71%). NTES has no false negatives as it classifies as untestable only the vertices that are directly untestable, disregarding dependencies.

RTM_{old}: By taking control flow dependencies into account, the false positive rate of RTM_{old} is lower than NTES, at the price of introducing a number of false negatives. False negatives appear because in some cases where the control flow does not propagate to all of the operations' dependencies, as we had assumed. Still, because of the assumptions

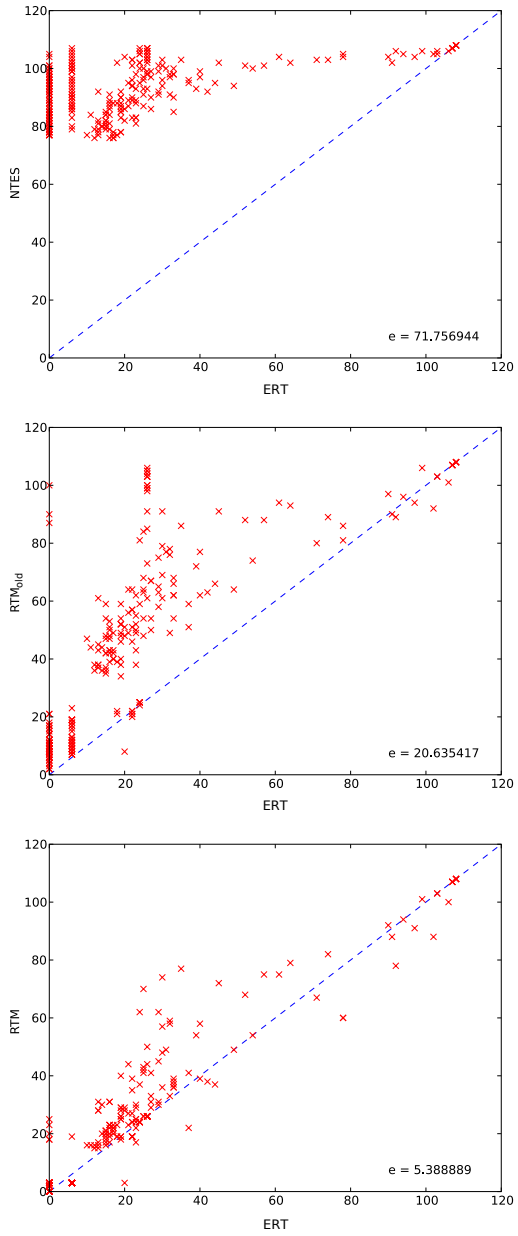

Figure 4. Accuracy of NTES, RTM$_{old}$ and RTM

that test interactions can start in any vertex, and that the test paths are independent, the number of false positives is considerable. The number of input vertices in WifiLounge is proportionally higher than for AISPlot. Hence, the number of false positives caused by this assumption is lower.

RTM: By taking input vertices into account, the amount of overestimation decreases dramatically for both systems. Still, the false positive rate is significant. Therefore, we conclude that assuming that paths are not dependent is not very reasonable and needs to be addressed in future work. The increase of false negatives makes more apparent the consequences of the assumption that control flow is always being transmitted to dependencies. The error caused by this assumption is augmented by the fact that it also applies to the input paths to reach the vertex being considered. Nevertheless, RTM correlates with the ERT, and did provide on average the minimal absolute error compared to the two other metrics.

After the theoretical and empirical validation of RTM, we will present application examples of it.

V. Application Examples

Two studies were performed on two component-based systems: (1) AISPlot, a system-of-systems taken from the maritime safety and security domain, and (2) WifiLounge, an airport's wireless access-point system. These two systems are representative of the two typical software architectures: the first system follows a data-flow organization, while the second one follows a client-server organization. These cases show that RTM can identify parts of a system with prohibitive runtime testing cost, and it can help choose optimal action points with the goal of improving the system's runtime testability. The $CIGs$ were obtained by static analysis of the code. The inter-component edges were obtained during runtime by reflection. The runtime testability and fix cost information c_i were derived based on test sensitivity information obtained from the design of each component, and the cost of deploying adequate test isolation measures. In order to keep the number of untestable vertices tractable, we considered that only operations in components whose state was too complex to duplicate (such as databases), or that caused external interactions (output components) would be considered untestable.

Table II shows the general characteristics for the architectures and graph models of the two systems used in our experiments, including number of components, vertices, and edges of each system.

	AISPlot	**WifiLounge**
Total components	31	9
Total vertices	86	159
Total edges	108	141

Table II
Characteristics of the systems

A. Example: AISPlot

In the first experiment we used a vessel tracking system taken from our industrial case study. It consists of a component-based system coming from the maritime safety and security domain. The architecture of the AISPlot system is shown in Figure 5. AISPlot is used to track the position of ships sailing a coastal area, detecting and managing potential dangerous situations. Messages are broadcast by ships

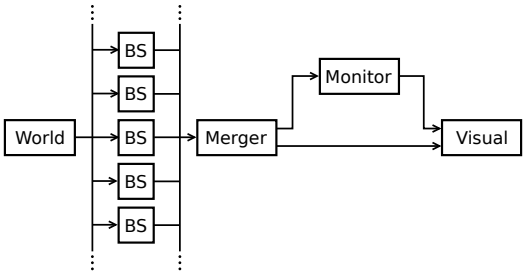

Figure 5. AISPlot Component Architecture

(represented in our experiment by the *World* component), and received by base stations (*BS* component) spread along the coast. Each message is relayed to a *Merger* component removing duplicates coming from different stations. Components interested in receiving status updates of ships, can subscribe to *Merger* to receive notifications. A *Monitor* component scans all messages looking for inconsistencies in the ship data, and another component, *Vis* shows all ships on a screen. Fig. 6 shows the CIG for AISPlot.

Cost	Proposed fix					Testability	
	v33	v34	v35	v36	v42	RTM	$rRTM$
0						12	0.140
1		×				17	0.198
2	×				×	21	0.244
3	×	×			×	26	0.302
4	×		×	×	×	81	0.942
5	×	×	×	×	×	86	1.000

Table III
TESTABILITY ANALYSIS FOR AISPLOT

Five *Vis* operations have testability issues (manually determined), displaying test ship positions and test warnings on the screen if not properly isolated. Table III shows that runtime testability is low. Only 14% of the vertices can be runtime tested. This poor RTM comes from the architecture of the system being organised as a pipeline, with the *Vis* component at the end, connecting almost all vertices to the five problematic vertices of the *Vis* component. We explored the possible combinations of isolation of any of these 5 vertices and computed the optimal improvement on RTM, assuming uniform cost of 1 to isolate an operation. Table III shows the best combination of isolation for each possible cost, × denoting the isolation of a vertex, and the RTM if these isolations were applied. The numbers suggest little gain in testability, as long as vertices v33, v35, v36 and v42 (corresponding respectively to operations in the visualiser for: new ships, status updates, disappearing ships, and warnings) are not made runtime testable together. This is caused by the topology of the graph: the four vertices appear at the end of the processing pipeline affecting the predecessor set of almost every vertex together. They must be fixed at once for any testability gain, leading to the jump at cost 4 for AISPlot in Figure 9 ($rRTM$ going from 0.302 to 0.942).

B. Example: WifiLounge

In a second experiment we diagnosed the runtime testability of a wireless hotspot at an airport lounge [19]. The component architecture of the system is depicted in Figure 8. Clients authenticate themselves as either business class passengers, loyalty program members, or prepaid service clients.

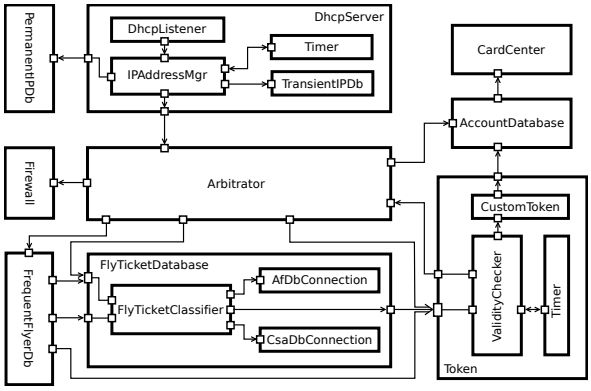

Figure 8. Wifi Lounge Component Architecture

When a client accesses the wifi network, a *DhcpListener* generates an event indicating the assigned IP address. All communication is blocked until authenticated. Business class clients are authenticated in the ticket databases of airlines. Frequent fliers are authenticated against the program's database, and the ticket databases for free access. Prepaid-clients must create an account in the system, linked to a credit card entry. After authentication, blocking is disabled and the connection can be used. Fig. 7 shows the CIG of *WifiLounge*.

Thirteen operations are runtime untestable, i.e., state modification operations of the *AccountDatabase*, *TransientIpDb* and *PermanentIpDb* components are considered runtime untestable because they act on databases. A withdraw operation of a *CardCenter* component is also not runtime testable because it uses a banking system outside our control. *Firewall* operations are also not runtime testable because this component is a front-end to a hardware element (the network), impossible to duplicate.

RTM is intermediate: 62% of the vertices can be runtime tested. This is much better than AISPlot, because the architecture is more "spread out" (compare both CIGs). There are runtime-untestable features, though they are not as interdependent as in AISPlot. We examined possible solutions improving RTM, displayed in Table IV, and shown in Fig. 9 (Airport Lounge). The number of vertices that have to be made runtime testable for a significant increase in RTM is much lower than for AISPlot. Two vertices (v14 and v18) cause the "biggest un-testability." The other vertices are not so problematic and the value of RTM grows more linearly with each vertex becoming runtime testable.

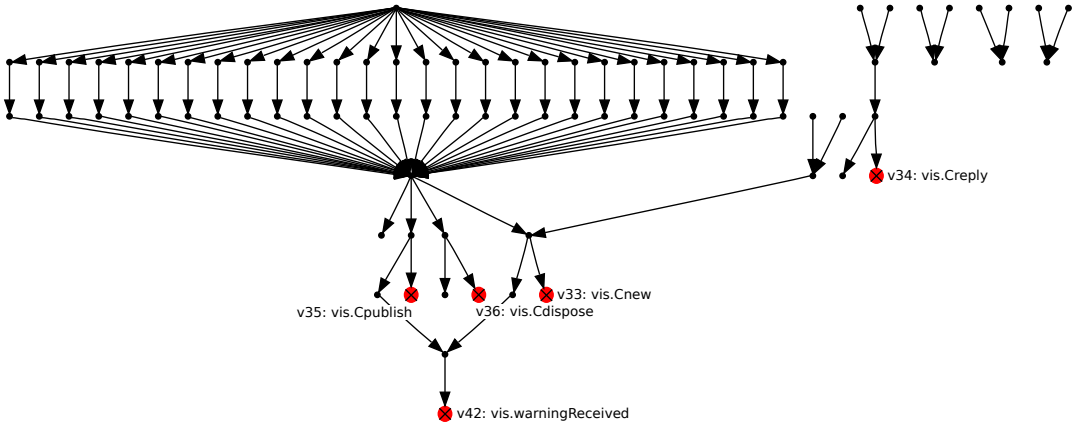

Figure 6. AISPlot Component Interaction Graph

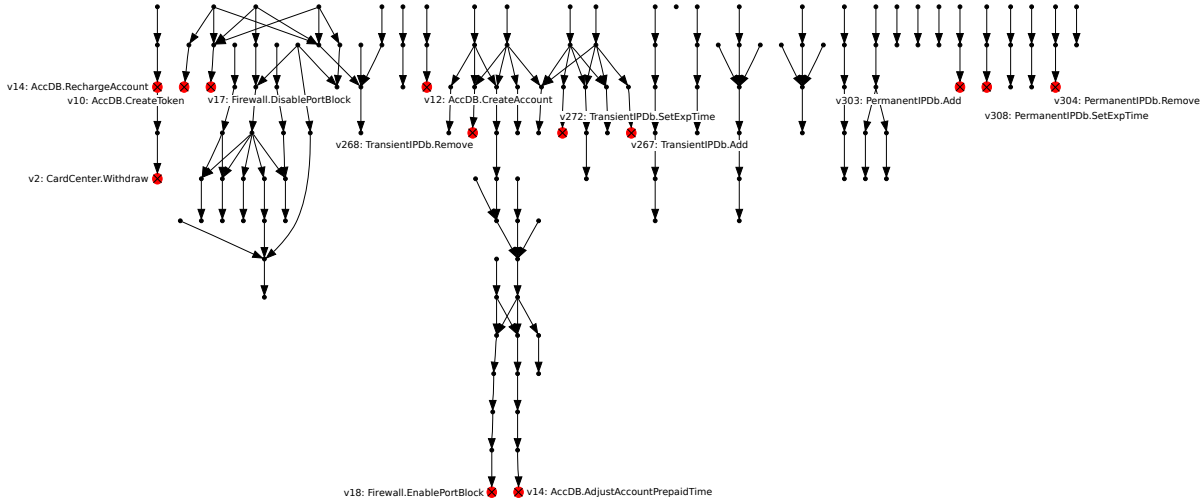

Figure 7. WifiLounge Component Interaction Graph

C. Discussion

The two cases demonstrate the value of RTM. By identifying operations causing inadmissible effects, we can predict the runtime untestable features, leading to an optimal action plan for runtime testable features. These techniques are applied before running test cases.

Because of the static model, RTM represents a pessimistic estimate, and we expect improvement by adding dynamic runtime information in the future, e.g., applying [20]. A high value, even if underestimated, is, nevertheless, a good indicator that the system is well prepared for runtime testing, and that the tests cover many system features. In future work, the value will be refined by providing dynamic information in the form of traversal probabilities, as proposed in the PPDG model presented in [20]. The design of the components on both systems was analysed in an effort to shed light on this issue. For instance, as we have seen in Section IV, for both *AISPlot* and *WifiLounge*, about 30 to 40% of the test cases were considered touching untestable operation by the RTM definition, although in reality they were not. This is due to the complexity of the control flow. Many exclusive branch choices are not represented in the static model.

An interesting issue is the relationship between RTM and defect coverage. Even though the relationship between test coverage and defect coverage is not clear [21], previous studies have shown a beneficial effect of test coverage on reliability [22], [23].

VI. Testability Optimisation

RTM analysis and action planning corresponds to the Knapsack problem [24], an NP-hard binary integer programming problem, which can be formulated as

$$maximise: \quad RTM$$
$$subject\ to: \quad \sum c(v_j) \cdot x_j \leq b, \quad x_j \in \{0,1\}$$

with b = maximum budget available, and x_j = decision of including vertex v_j in the action plan. In this section,

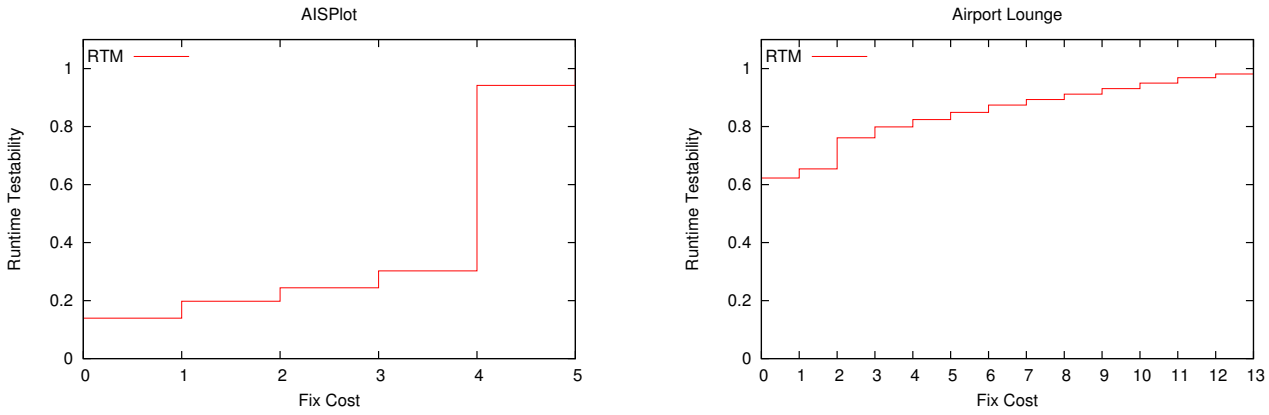

Figure 9. Optimal improvement of RTM vs. Fix cost

Cost	Proposed fix													Testability	
	v2	v10	v12	v13	v14	v17	v18	v267	v268	v272	v303	v304	v308	RTM	$rRTM$
0														99	0.623
1						×								103	0.648
2					×		×							121	0.761
3					×		×		×					127	0.799
4					×	×	×		×					131	0.824
5					×	×	×	×	×					135	0.849
6					×	×	×	×	×	×				139	0.874
7		×			×	×	×	×	×	×				142	0.893
8		×	×		×	×	×	×	×	×				145	0.912
9	×	×		×	×	×	×	×	×	×				147	0.925
10	×	×	×	×	×	×	×	×	×	×				150	0.943
11	×	×	×	×	×	×	×	×	×	×	×			153	0.962
12	×	×	×	×	×	×	×	×	×	×	×	×		156	0.981
13	×	×	×	×	×	×	×	×	×	×	×	×	×	159	1.000

Table IV
TESTABILITY ANALYSIS OF AIRPORT LOUNGE

we present a way for an approximate action plan using the greedy heuristic method according to Algorithm 1, in which CIG is the interaction graph, U is the set of untestable vertices, and $H(v)$ a heuristic function to be used. For each pass of the loop, the algorithm selects the vertex in U with the highest heuristic rank, and removes it from the set of untestable vertices. The rank is updated on each pass.

Algorithm 1 Greedy Approximate Planning

```
function FIXACTIONPLAN(CIG, U, H(v))
    Sol ← ∅                          ▷ List to hold the solution
    while U ≠ ∅ do
        v ← FINDMAX(U, H(v))
        APPEND(Sol, v)
        REMOVE(U, v)
    return Sol
```

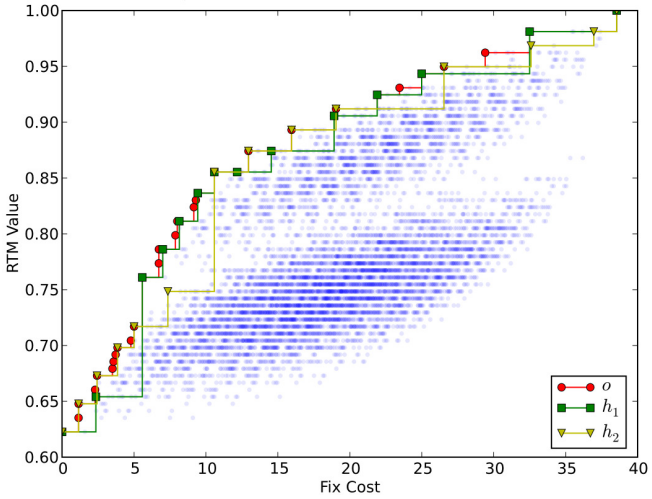

Figure 10. Optimal and heuristic RTM optimisations

The method relies on heuristics that benefit from partial knowledge about the structure of the solution space of the problem. To motivate our heuristic approach, we analyse the properties of the RTM-cost combination space shown in Fig. 10. The dot-clouds show the structure and distribution of all the possible solutions for the vertex and context-dependence RTM optimisation problems of the *WifiLounge* system. On a system where the cost of fixing any vertex is uniform, and all the uncoverable vertices or paths come from only one untestable vertex, there would be only one cloud. However, we identified two interesting characteristics of the inputs affecting the structure of the solution space.

First, in most systems multiple untestable vertices will participate in the same uncoverable elements. This is he case for both examples. If a group of untestable vertices participates together in many un-coverable features, the

solution cloud will cast a "shadow" on the RTM axis, i.e., any solution that includes those vertices will get a better testability. Second, vertices with exceptionally high cost will shift any solution that includes them towards the right in the cost axis, causing a separate cloud to appear. In this case, any solution that contains them will get a cost increase, due to space concerns not shown in the plots. An example of the first characteristic is in Figure 10, where the upper cloud corresponds to all the solutions that include vertices v14 and v18. We used the knowledge about these two situations to define heuristics to be used in Algorithm 1, based on the idea that dependent vertices are only useful if they are all part of the solution and expensive vertices should be avoided unless necessary.

A. Heuristics

First, we consider a pessimistic heuristic. It ranks higher the vertices with the highest gain on testability. The count is divided by the cost to penalise expensive nodes:

$$h_{pessimistic}(v_i) = \frac{1}{c_i}(RTM_{v_i} - RTM) \tag{8}$$

with RTM_{v_i} = RTM after the cost for vertex v_i was spent. Given the pessimistic nature of this heuristic, we expect this heuristic to perform well for low budgets, and poorly for higher ones. It should be noted that we can define this heuristic because RTM was proven to have ratio scale (cf Section III).

The second heuristic is optimistic. It ranks higher the vertices that appear in the highest number of P sets, i.e., the vertices that will fix the most uncoverable vertices assuming they only depend on the vertex being ranked. This value is also divided by the cost to penalise expensive nodes over cheaper ones:

$$h_{optimistic}(v_i) = \frac{1}{c_i}|\{v_j \mid v_i \in S_{v_j}\}| \tag{9}$$

By ignoring the fact that an uncoverable vertex may be caused by more than one untestable vertex, and that the vertex may not be reachable through a testable path, this second heuristic will take very optimistic decisions on the first passes, affecting the quality of results for proportionally low budgets, but yields a better performance for higher ones. Although this heuristic ignores uncoverable elements that depend on multiple vertices of U, if two vertices appear together in many P_i sets, their ranks will be similar and will be chosen one after the other.

Fig. 10 shows the performance for both heuristics (h_1 & h_2) for the *WifiLounge* (compared to the optimal solution o, obtained by exhaustive search). The steps in the optimal solution are not incremental and the action plans at each step could be completely different. The optimistic ranking skips many low-cost solutions (with curve much lower than the optimum), while the pessimistic heuristic is more precise for low cost, but completely misses good solutions with higher budgets. These shortcoming may be addressed through combining both heuristic rankings and taking the best results of both. However, the steps in the solution will not be incremental if the solutions intersect with each other (as in Figure 10).

B. Computational Complexity and Error

The time complexity of the *action plan* function depends on the complexity of the heuristic in Algorithm 1. As in each pass there is one less vertex in U, the H function is evaluated $|U|, |U|-1, \ldots, 1$ times while searching for the maximum. In total, it is evaluated $\frac{|U|^2}{2}$ times. Both heuristics perform a sum depending on the number of vertices. Hence, the complexity of the *action plan* function is $O(|V| \cdot |U|^2)$, i.e., polynomial.

Although polynomial complexity is much more appealing than the $O(2^{|U|})$ complexity of the exhaustive search, the approximation error must be considered. Experiments were conducted to evaluate the approximation error of our heuristics. The graph structures of *AISPlot* and *WifiLounge* were used, randomly altering the untestable vertices, and the preparation cost information (chosen according to a Pareto distribution). The plot in Figure 11 shows the evolution of the relative average approximation error of RTM for our heuristics as a function of the number of untestable operations $|U|$. The optimal solution function is obtained by exhaustive search.

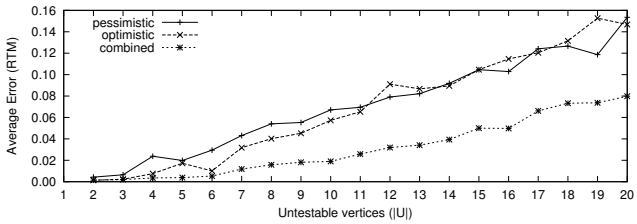

Figure 11. Performance of the approximate algorithms vs. the optimal

The average error incurred by our heuristics is very low w.r.t. the processing time for their calculation. It is notable that the error has an increasing trend, and the pessimistic and optimistic heuristics are similar. Combining the rankings created by both heuristics, choosing the maximum of either solution, reduces the error while maintaining the low computational complexity. This can be seen in the $combined$ error plot in Figure 11.

VII. Implementation

In order to further validate the applicability of the RTM in software projects, we have integrated the measurement of this metric into our component framework Atlas[1]. For a component-based system, in order to build the CIG, the

[1] http://swerl.tudelft.nl/bin/view/Main/Atlas

graph from which the metric is computed, three types of information are necessary:

- The external connections between components,
- The internal connections inside a component,
- The list of test-sensitive operations.

Within the Atlas framework, the connection between component interfaces is explicit, written in the architecture description file by the developer. In component frameworks where these connections are implicit and created dynamically, they can be still be determined at runtime automatically, as in such infrastructure the framework is always in charge of the connection of the components. Per definition, a component is seen by the framework as a black-box, and therefore the information on the internal connections cannot be directly determined. In our case, we have extended the architecture declaration language to permit encoding such information. The encoding is a simple list of tuples indicating which input interface might call which output interface. Nevertheless, this doesn't mean this cannot be automated. For instance, for the generation of this information for the two example systems presented previously in Section V, static analysis was used to discover most of the connections. A quick review by the developer is sufficient to correct the few missing connections. Finally, the information concerning the test sensitivity is the most delicate one to obtain. It is best to consider by default every operation, which is in this context an input interface, as test-sensitive. Then a review by the developer is necessary to define one-by-one which of the operation cannot cause any interference when tested at runtime. In our case, the architecture description language was extended to contain this information provided by the developer.

Figure 12 shows a screenshot of our framework after computation of the RTM. The user interface shows the components in the system on the left. At the request of the user, the CIG is displayed in the right pane. On the very right side, stands the rRTM for the system.

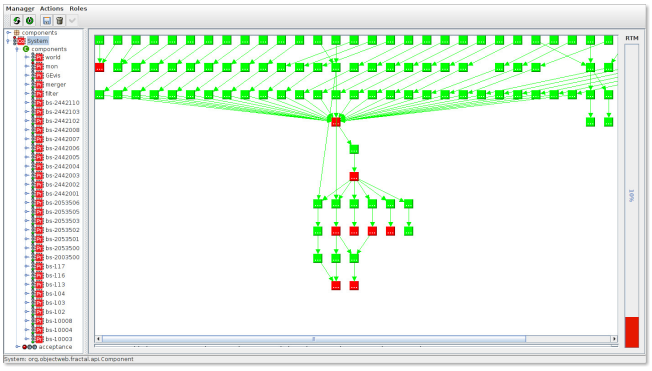

Figure 12. User interface of our component framework displaying the CIG and rRTM of a system.

VIII. Related work

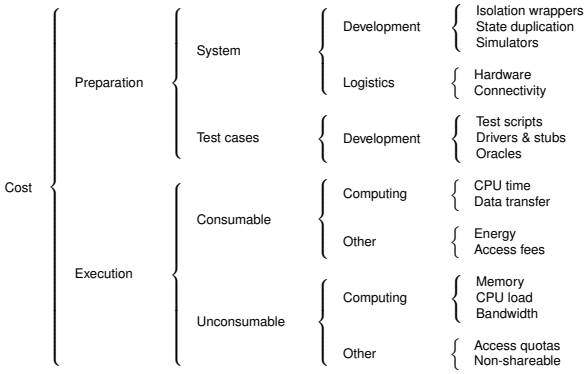

Figure 13. Taxonomy of Testing Costs

A number of research approaches have addressed testability from different angles. The cost of testing is actually complex and decomposed into multiple different types of costs.

Figure 13 presents a taxonomy of testing costs. On the upper branch of the diagram, costs associated with the preparation can be found. These costs correspond to testability improvement efforts which are performed before the tests are carried out. Test interferences are critical because, in the worst case, runtime tests will affect the system's environment in ways that are unexpected and difficult to control, or impossible to recover from, e.g., firing a missile while testing part of a combat system. Therefore the preparation costs include, for example, the adaptation of the system to support isolation and state duplication, or the development of simulators for some components. Preparation costs also include logistic provisioning, such as increasing the capacity of the system by using hardware capable of handling the operational and testing loads simultaneously. Of course, the development of the tests themselves, under the form of test cases, is also a large contribution to the preparation costs.

On the lower branch of the diagram costs associated to the execution of the test case on the system under test. Consumable costs are used up by executing tests, the most traditional example being time. Non-consumable costs, on the other hand, are "on loan" during the execution of the test, and returned when the test finishes. For example, CPU load is bounded (to 100%) and will not only depend on the quantity of tests applied but only on their "concentration": the CPU load might decreased simply by spreading tests over a longer period of time. These non-consumable costs are the one that can affect the non-functional requirements of the system when employing runtime testing.

Runtime testability has not been considered in depth so far. In such context, the system preparation cost for testing has a strong importance. To the best of our knowledge, our paper is the first to (1) define a measurement (RTM) in a

way that is conductive to an estimation of runtime testability, (2) consider testability improvement planning in terms of a testability/cost optimisation problem, and (3) to present a near-optimal, low-cost heuristic algorithm to compute the testability optimization plan.

Traditionally, test cost minimisation has considered only execution cost. To reduce the consumable costs test effort minimisation algorithms have been proposed, both for test time and coverage [8], [9], [10], [11] or test time and reliability [25], [26]. Test sensitivity and isolation, are introduced by Brenner et al. [4] to reduce the non-consumable costs, however no mention to nor relation with the concept of runtime testability were presented. On the same topic, Suliman et al. [27] discuss several test execution and sensitivity scenarios, for which different isolation strategies are advised. These two works form the base for our initial approach to runtime testability, presented in [12], [13], and extended and more thoroughly evaluated in this paper, which is an extended version of [1]. The factors that affect runtime testability cross-cut those in Binder's Testability [28] model, as well as those in Gao's component-based adaptation [29].

Other testability-related approaches have focused on modeling statistically which characteristics of the source code of the system are more prone to uncovering faults [6], [7] for amplifying reliability information [5], [30]. Preparation cost, understood as the compilation time overhead caused by the number of dependencies needed to test any other component, was addressed in [31]. They proposes a measurement of testability from the point of view the static structure of the system, to assess the maintainability of the system. Our approach is similar in that runtime testability is influenced by the structure of the system under consideration.

IX. Conclusions and Future Work

The amount of runtime testing that can be performed on a system is limited by the characteristics of the system, its components, and the test cases themselves.

In this paper, we have studied RTM, a cost-based measurement for the runtime testability of a component-based system, which provides valuable information to test engineers about the system, independently of the actual test cases that will be run. RTM has been validated from a theoretical point of view that it conforms to the notion of measurement, and that it can be used to *rate* systems. An empirical validation has shown that it provides with relatively good accuracy prediction of the actual runtime testability. Furthermore, we have introduced an approach to the improvement of the system's runtime testability in terms of a testability/cost optimisation problem, which allows system engineers to elaborate an action plan to direct the implementation of test isolation techniques with the goal of increasing the runtime testability of the system in an optimal way. We have provided a low-cost approximation algorithm which computes near-optimal improvement plans, reducing significantly the computation time. This algorithm is well suited for usage in an interactive tool, enabling system engineers to receive real-time feedback about the system they are integrating and testing at runtime.

Future work towards extending the impact cost model with values in the real domain instead of a boolean flag will be carried out. This work could benefit from the test cost estimation and reduction techniques cited in the related work, and be used to devise a runtime-test generation and prioritisation algorithm that attempts to achieve the maximum coverage with the minimum impact for the system. Moreover, because the RTM as obtained by our method is a lower bound, further work will encompass an effort to improve its accuracy, by enriching the model with dynamic information in the form of edge traversal probabilities. Finally, additional empirical evaluation using industrial cases and synthetic systems will be carried out in order to explore further the relationship between RTM and defect coverage and reliability.

Acknowledgement

This work is part of the ESI Poseidon project, partially supported by the Dutch Ministry of Economic Affairs under the BSIK03021 program.

References

[1] A. Gonzalez-Sanchez, É. Piel, H.-G. Gross, and A. van Gemund, "Runtime testability in dynamic high-availability component-based systems," in *The Second International Conference on Advances in System Testing and Validation Lifecycle*, Nice, France, Aug. 2010.

[2] D. Brenner, C. Atkinson, O. Hummel, and D. Stoll, "Strategies for the run-time testing of third party web services," in *SOCA '07: Proceedings of the IEEE International Conference on Service-Oriented Computing and Applications*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 114–121.

[3] A. González, É. Piel, H.-G. Gross, and M. Glandrup, "Testing challenges of maritime safety and security systems-of-systems," in *Testing: Academic and Industry Conference - Practice And Research Techniques*. Windsor, United Kingdom: IEEE Computer Society, Aug. 2008, pp. 35–39.

[4] D. Brenner, C. Atkinson, R. Malaka, M. Merdes, B. Paech, and D. Suliman, "Reducing verification effort in component-based software engineering through built-in testing," *Information Systems Frontiers*, vol. 9, no. 2-3, pp. 151–162, 2007.

[5] A. Bertolino and L. Strigini, "Using testability measures for dependability assessment," in *ICSE '95: Proceedings of the 17th international conference on Software engineering*. New York, NY, USA: ACM, 1995, pp. 61–70.

[6] R. S. Freedman, "Testability of software components," *IEEE Transactions on Software Engineering*, vol. 17, no. 6, pp. 553–564, 1991.

[7] J. Voas, L. Morrel, and K. Miller, "Predicting where faults can hide from testing," *IEEE Software*, vol. 8, no. 2, pp. 41–48, 1991.

[8] S. Elbaum, A. Malishevsky, and G. Rothermel, "Test case prioritization: A family of empirical studies," *IEEE Transactions on Software Engineering*, vol. 28, pp. 159–182, 2002.

[9] Z. Li, M. Harman, and R. M. Hierons, "Search algorithms for regression test case prioritization," *IEEE Transactions on Software Engineering*, vol. 33, no. 4, pp. 225–237, 2007.

[10] A. M. Smith and G. M. Kapfhammer, "An empirical study of incorporating cost into test suite reduction and prioritization," in *24th Annual ACM Symposium on Applied Computing (SAC'09)*. ACM Press, Mar. 2009, pp. 461–467.

[11] Y. Yu, J. A. Jones, and M. J. Harrold, "An empirical study of the effects of test-suite reduction on fault localization," in *International Conference on Software Engineering (ICSE 2008)*, Leipzig, Germany, May 2008, pp. 201–210.

[12] A. Gonzalez-Sanchez, É. Piel, and H.-G. Gross, "RiTMO: A method for runtime testability measurement and optimisation," in *Quality Software, 9th International Conference on*. Jeju, South Korea: IEEE Reliability Society, Aug. 2009.

[13] A. Gonzalez-Sanchez, É. Piel, H.-G. Gross, and A. J. van Gemund, "Minimising the preparation cost of runtime testing based on testability metrics," in *34th IEEE Computer Software and Applications Conference*, Seoul, South Korea, Jul. 2010.

[14] J. Radatz, "IEEE standard glossary of software engineering terminology," *IEEE Std 610.12-1990*, Sep. 1990. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=159342

[15] A. González, E. Piel, and H.-G. Gross, "A model for the measurement of the runtime testability of component-based systems," in *Software Testing Verification and Validation Workshop, IEEE International Conference on*. Denver, CO, USA: IEEE Computer Society, 2009, pp. 19–28.

[16] Y. Wu, D. Pan, and M.-H. Chen, "Techniques for testing component-based software," in *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems*. Los Alamitos, CA, USA: IEEE Computer Society, 2001, pp. 222–232.

[17] M. Shepperd and D. Ince, *Derivation and Validation of Software Metrics*. Oxford University Press, 1993.

[18] H. Zuse, *A Framework of software measurement*. Hawthorne, NJ, USA: Walter de Gruyter & Co., 1997.

[19] T. Bures. (2011, Jun.) Fractal BPC demo. [Online]. Available: http://fractal.ow2.org/fractalbpc/index.html

[20] G. K. Baah, A. Podgurski, and M. J. Harrold, "The probabilistic program dependence graph and its application to fault diagnosis," in *International Symposium on Software Testing and Analysis (ISSTA 2008)*, Seattle, Washington, Jul. 2008, pp. 189–200.

[21] L. Briand and D. Pfahl, "Using simulation for assessing the real impact of test coverage on defect coverage," in *Proceedings of the 10th International Symposium on Software Reliability Engineering*, 1999, pp. 148–157.

[22] X. Cai and M. R. Lyu, "Software reliability modeling with test coverage: Experimentation and measurement with a fault-tolerant software project," in *Proceedings of the 18th IEEE International Symposium on Software Reliability*. Washington, DC, USA: IEEE Computer, 2007, pp. 17–26.

[23] M. A. Vouk, "Using reliability models during testing with non-operational profiles," in *Proceedings of the 2nd Bellcore/Purdue workshop on issues in Software Reliability Estimation*, 1992, pp. 103–111.

[24] R. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations*, R. Miller and J. Thatcher, Eds. Plenum Press, 1972, pp. 85–103.

[25] C. Y. Huang and M. R. Lyu, "Optimal release time for software systems considering cost, testing-effort, and test efficiency," *IEEE Transactions on Reliability*, vol. 54, no. 4, pp. 583–591, 2005.

[26] K. Okumoto and A. Goel, "Optimum release time for software systems based on reliability and cost criteria," *Journal of Systems and Software*, vol. 1, pp. 315–318, 1980.

[27] D. Suliman, B. Paech, L. Borner, C. Atkinson, D. Brenner, M. Merdes, and R. Malaka, "The MORABIT approach to runtime component testing," in *30th Annual International Computer Software and Applications Conference*, Sep. 2006, pp. 171–176.

[28] R. V. Binder, "Design for testability in object-oriented systems," *Communications of the ACM*, vol. 37, no. 9, pp. 87–101, 1994.

[29] J. Gao and M.-C. Shih, "A component testability model for verification and measurement," in *COMPSAC '05: Proceedings of the 29th Annual International Computer Software and Applications Conference*, vol. 2. Washington, DC, USA: IEEE Computer Society, 2005, pp. 211–218.

[30] D. Hamlet and J. Voas, "Faults on its sleeve: amplifying software reliability testing," *SIGSOFT Software Engineering Notes*, vol. 18, no. 3, pp. 89–98, 1993.

[31] S. Jungmayr, "Identifying test-critical dependencies," in *ICSM '02: Proceedings of the International Conference on Software Maintenance (ICSM'02)*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 404–413.

The Economic Importance of Business Software Systems Development and Enhancement Projects Functional Assessment

Beata Czarnacka-Chrobot
Department of Business Informatics
Warsaw School of Economics
Warsaw, Poland
e-mail: bczarn@sgh.waw.pl

Abstract—**Execution of Business Software Systems (BSS) Development and Enhancement Projects (D&EP) encounters many problems, leading to the high scale of their failure, which then is reflected in considerable financial losses. One of the fundamental causes of such projects' low effectiveness are improperly derived estimates for their costs and time. In their case, the budget and time frame are determined by the effort being spent on activities needed to deliver product that would be meeting client's requirements. Meanwhile, objective and reliable effort estimation still appears to be a great challenge, what in the author's opinion is caused by effort estimation based on resources, while such planning activity should base on the required software product size, which determines work effort. Estimation of BSS size requires using of the suitable software size measure, which has been sought for several decades now. What's more, it is worth using the capabilities offered by such measure for the BSS D&EP assessment from the perspective being critical to a client, that is from functional perspective. Thus this paper analyses capabilities, being significant from the economic point of view, of taking advantage of suitable approach to the BSS size measurement, what should contribute to the better understanding of the importance of this issue, still being underestimated by business managers – as in the subject literature this issue is usually considered from the technical point of view. Meanwhile, suitable BSS size measurement should constitute the basis for rational activities and business decisions not only for providers, but also for clients needs.**

Keywords-business software systems development and enhancement projects; effectiveness; software size measures; functional size measurement; functional assessment; Software projects Functional Assessment Model (SoftFAM)

I. SCALE OF FAILURES IN THE BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS EXECUTION

In practice, the execution of software Development and Enhancement Projects (D&EP), particularly those delivering Business Software Systems (BSS) as a product, encounters many problems, which makes fulfilling of client requirements still appear a big challenge for companies dealing with this kind of business (see also [1]). This may be proved by the unsatisfactory effectiveness of such projects, revealed by numerous analyses, which manifests itself in the high scale of their failure.

The Standish Group, the US institution providing research reports on this issue from 15 years, estimates that now only 32% of application D&EP worldwide turn out successful while products delivered as a result of nearly 45% of them lack on average 32% of the required functions and features, the planned time of product delivery is exceeded by nearly 80% on average and the estimated budget - by approx. 55% on average [2]. Also, it is worth mentioning the research carried out by government agencies in the USA indicating that 60% of software systems development projects overrun the planned completion time, 50% of these projects overrun the estimated costs while in the case of 46% of them the delivered products turn out useless [3]. Similar – as to the general conclusion – data result from the analysis of IT projects being accomplished in Poland, which was carried out by M. Dyczkowski, indicating that in 2006-2007 approx. 48% of such projects went over the planned completion time while approx. 40% exceeded the estimated budget [4].

Analyses by T.C. Jones plainly indicate that those software D&EP, which are aimed at delivery of business software systems, have the lowest chance to succeed [5]. The Panorama Consulting Group, when investigating in their 2008 study the effectiveness of ERP (Enterprise Resource Planning) systems projects being accomplished worldwide revealed that 93% of them were completed after the scheduled time while as many as 68% among them were considerably delayed comparing to the expected completion time [6]. Merely 7% of the surveyed ERP projects were accomplished as planned. Comparison of actual versus planned expenses has revealed that as many as 65% of such projects overran the planned budget. Only 13% of the respondents expressed high satisfaction with the functionality implemented in final product while in merely every fifth company at least 50% of the expected benefits from its implementation were said to be achieved. Meanwhile (see also [4][7]):

- BSS are one of the fundamental IT application areas.
- BSS development or enhancement often constitutes serious investment undertaking.
- In practice, COTS (Commercial-Off-The-Shelf) BSS rarely happen to be fully tailored to the particular client business requirements therefore their customisation appears vital.

- Rational *ex ante* and *ex post* valuation of unique (at least partially) BSS, being of key significance to clients, encounters serious problems in practice.
- From the provider's perspective, the discussed type of IT projects is particularly difficult in terms of management, which basically results in their exceptionally low effectiveness as compared to other types of IT projects.

The paper is structured as follows: in Section 2 the author presents the selected results of studies concerning losses caused by the especially low effectiveness of BSS D&EP execution and points out main factors of BSS D&EP effectiveness. In Section 3 different BSS size measures are compared, while in Section 4 the concept and methods of BSS functional size measurement are analysed. Section 5 is devoted to the presentation of author's own model dedicated to BSS D&EP functional assessment against a background of existing related methodologies and along with the main conclusions coming from its verification and comparison to those methodologies. In Section 6 the main results of author's own study on the usage of functional size measurement methods by Polish BSS providers are pointed out. Finally, in Section 7 the author draws conclusions and some open lines about future work on functional approach to the BSS D&EP assessment from the economic point of view.

II. LOSSES CAUSED BY THE LOW EFFECTIVENESS OF BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS EXECUTION

Low effectiveness of BSS D&EP execution leads to the substantial financial losses, on a worldwide scale estimated to be hundreds of billions of dollars yearly, sometimes making even more than half the funds being invested in such projects. The Standish Group estimates that these losses – excluding losses caused by business opportunities lost by clients, providers losing credibility or legal repercussions – range, depending on the year considered, from approx. 20% to even 55% of the costs assigned for the execution of the analysed projects types (see e.g., [8][9]). On the other hand, analyses of The Economist Intelligence Unit, which studied the consequences of BSS D&EP delay indicate that there is strong correlation between delays in delivery of software products and services and decrease in profitability of a company therefore failures of BSS D&EP, resulting in delays in making new product and services available and in decreasing the expected income represent threat also to the company's business activity [10]. Meanwhile,"The costs of these (...) overruns are just the tip of the proverbial iceberg. The lost opportunity costs are not measurable, but could easily be in the trillions of dollars. [For instance - B.C.C.] the failure to produce reliable software to handle baggage at the new Denver airport is costing the city $1,1 million per day." [11].

If direct losses caused by abandoning the BSS D&EP result from erroneous allocation of financial means, usually being not retrievable, in the case of overrunning the estimated time and/or costs, however, they may result from delay in gaining the planned return on investment as well as from decreasing it (necessity to invest additional funds and/or cutting on profits due to the overrunning of execution time and/or delivery of product incompatible with requirements).

According to the Standish Group analyses, yearly spendings on application software D&EP in the USA range from approx. 250 to approx. 350 billion USD. In this type of projects, average yearly cost of development works alone ranges from approx. 0,4 to approx. 1,6 million USD, what indicates that they are usually serious investment undertakings. Spendings on such projects may considerably exceed the expense of building offices occupied by companies commissioning them, and in extreme cases, even 50-storey skyscraper, roofed football stadium, or cruising ship with a displacement of 70.000 tons [12]. Yet quite often client spends these sums without supporting their decision on getting engaged in such investment by proper analysis of the costs, based on the rational, sufficiently objective and reliable grounds. The above situation manifests itself in the difference in costs spent by various organizations on similar applications that may be even fifteen fold [13].

The above unequivocally implies a significant need to rationalize practical activities and business decisions made with regard to BSS D&EP, which is only possible when taking into account factors showing influence on this effectiveness. Author's analysis, which concerned numerous studies on factors of BSS D&EP effectiveness, available in the subject literature, leads to the conclusion that among fundamental factors are:

1) Proper project management, including: realistic planning, with particular consideration given to the reliable and objective estimates for key project attributes (work effort, execution time and cost), and proper project scope management, above all consisting in undertaking small projects, that is projects whose product is characterised by relatively small size. Both these factors require product size measurement.

2) Authentic involvement of client in the project – both users and managers. Thus product size measurement should be carried out by taking into consideration mainly the perspective of the client of BSS being developed, that is with the use of product size units that are of high significance to him.

Therefore if fundamental opportunity to increase the chance for effective execution of the discussed types of projects and to decrease the losses caused by low effectiveness lies in accurate estimates of their key attributes, in undertaking small projects and in client's involvement then what appears to be *significant factor of BSS D&EP success is objective and reliable measurement of their product size*, with particular consideration given to client's perspective. "Measurement of software size (...) is as important to a software professional as measurement of a building (…) is to a building contractor. All other derived data, including effort to deliver a software project, delivery schedule, and cost of the project, are based on one of its major input elements: software size." [14, p. 149].

III. BUSINESS SOFTWARE SYSTEMS SIZE MEASURES

One of the fundamental causes of low BSS D&EP success rate are improperly derived estimates for their costs and time. In the case of such projects the budget and time frame are determined by the effort being spent on activities needed to deliver product, which would meet client's requirements. However, sufficiently objective and reliable BSS D&EP effort estimation still appears to be a great challenge to the software engineering. In the author's opinion the main reason for this problem is effort estimation made on the basis of resources whereas such planning activity should ground on the required software product size, which determines the work effort (see Figure 1).

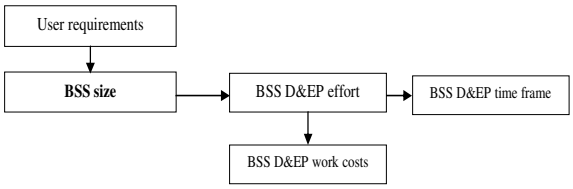

Figure 1. Simplified model of dependencies between BSS D&EP key attributes and the size of project product.

Source: Author's own study.

Basic approaches to the size measurement of every software product may be reduced to perceiving it from the perspective of (see also [7]):

- Length of programmes, measured by the number of the so-called programming (volume) units. These units most of all include source lines of code, but number of commands, number of machine language instructions are also taken into account. However, these units measure neither size of the programmes nor their complexity but only the attribute of "programme length" yet thus far these are them that in practice have been employed most often with regard to the software size [14, p. 149].
- Software construction complexity, measured in the so-called construction complexity units. Most of hundreds of such measures having been proposed are limited to the programme code yet currently these units are used mainly in the form of object points [14, pp. 155-156]. These points are assigned to the construction elements of software (screens, reports, software modules) depending on the level of their complexity.
- Functionality of software product, expressed in the so-called functionality units. They most of all include function points, but also variants based on them such as: full function points, feature points, or use case points. These points are assigned to the functional elements of software (functions and data needed to complete them) depending on the level of their complexity – not to the construction elements as it was the case of object points.

Synthetic comparison of various software size measures against a background of key requirements set for such measures were presented in Table 1.

TABLE I. SYNTHETIC COMPARISON OF SOFTWARE SIZE MEASURES

Requirement towards measures	Programming units	Construction complexity units	Functionality units
Unequivocalness of definition	Freedom in formulating definitions (differences as big as even 5:1)	Depending on the method	In methods normalized by ISO/IEC
Possibility to make reliable prognosis on the size relatively early in the life cycle	Possibility to calculate programme length only for the existing code	None – with regard to programming units and object points	As early as at the stage of requirements specification
Base for the reliable evaluation of the all phases work effort	Final programme length does not fully reflect the whole work done	Final software size does not fully reflect the whole work done	Relatively high reliability as early as at the stage of requirements specification
Software size being independent of the technology employed	Programme length determined by the language employed	Size being dependent on the technology employed	Size depends on functional user requirements
Possibility to compare software written in different languages	Lack of such direct possibility	Lack of such direct possibility	Size doesn't depend on the language used
Measuring size in units being of significance to a client	No significance to a client	Secondary significance to a client	Measurement from the point of view of a client
Possibility to compare delivered size vs. required size	Inability to make reliable prognosis	Inability to make reliable prognosis	Thanks to the possibility of making reliable prognosis
Possibility to measure all software categories	Yes	Depending on the method	Depending on the method
Easiness of use	Yes	No	No

Source: Author's own study.

IV. BUSINESS SOFTWARE SYSTEMS FUNCTIONAL SIZE MEASUREMENT

The right measure of software size has been sought out for several decades now. Many years' verification of various approaches showed that what for now deserves standardization is just the concept of software size measurement based on its functionality – being an attribute of first priority to the client. Due to the empirically confirmed effectiveness of such approach, it was in the last years normalized by the ISO (International Organization for Standardization) and IEC (International Electrotechnical Commission), and turned into the six-part international standard ISO/IEC 14143 for the so-called software Functional Size Measurement (FSM) [15].

Details displayed in Table 1 clearly indicate the reasons why functionality units were recognised as the most appropriate measure of software size not only by the

ISO/IEC but also, among others, by Gartner Group [16] as well as by International Software Benchmarking Standards Group (ISBSG) [17]. They show no limits being characteristic of programming units and construction complexity units – although one may have reservations as to their versatility and relatively high complexity of the methods based on them. However, it is hard to expect that the method of measurement of software products, by nature being complicated, would be effective yet simple.

The set of rules for software FSM enclosed in the ISO/IEC 14143 norm provides key definitions, characteristics and requirements for FSM, and defines Functional Size Measurement Method (FSMM) as a specific FSM implementation defined by a set of rules, which conforms to the mandatory features of such measurement. The first part of this standard defines also indispensable principles upon which the FSMM should be based – fundamental one is the definition of functional size, which is understood as "a size of the software derived by quantifying the functional user requirements", while The Functional User Requirements (FUR) "represent the user practices and procedures, that the software must perform to fulfil the user's needs" [15, Part 1].

After about 30 years of improving various FSM techniques five of them (out of over 20) have been now acknowledged by the ISO/IEC as conforming to the rules laid down in the ISO/IEC 14143 standard, namely:

- International Function Point Users Group (IFPUG) method [18].
- Mark II (MkII) function point method proposed by the United Kingdom Software Metrics Association (UKSMA) [19].
- Netherlands Software Metrics Association (NESMA) function point method [20].
- Common Software Measurement International Consortium (COSMIC) full function points method [21].
- FSM method developed by the Finnish Software Measurement Association (FiSMA) [22].

The FSMM standardized by the ISO/IEC differ in terms of software measurement capabilities with regard to different software classes (functional domains), but all of them are adequate for business software systems (see Table 2 and [23]).

Functional size measurement of BSS supports first of all [15, Part 6]:

- BSS D&EP management by enabling: to make early prognosis on resources necessary for project execution, to monitor progress in project execution, to manage the changes in the required BSS size, to determine degree to which the supplied product meets functional user requirements, as well as to make post-execution project analysis and compare its attributes with other projects.
- BSS performance management by: BSS development, enhancement and maintenance productivity management, quality management, organizational processes maturity and capability management, determining organizational BSS asset value in order to estimate cost of its potential replacement, reengineering, or outsourcing, making prognosis on budget necessary to maintain BSS, as well as BSS supply contracts management.

TABLE II. THE ISO/IEC STANDARDS FOR THE SOFTWARE FSMM VERSUS FUNCTIONAL DOMAINS

FSMM	Functional domains specified in the norm	Constraints indicated in the norm
ISO/IEC 20926 - IFPUG method	All software classes	None
ISO/IEC 20968 - UKSMA method	For any type of software provided that the so-called logical transactions may be identified in it (the rules were developed as intended for business software).	The rules support neither complex algorithms characteristic of scientific and engineering software nor the real-time systems.
ISO/IEC 24570 – NESMA method	All software classes	None
ISO/IEC 19761 - COSMIC-FFP method	- Data-driven systems, e.g., business applications for banking, insurance, accounting, personnel management - Real-time systems (time-driven systems), e.g., telephone exchange systems, embedded software, process control, operation systems - Hybrid solutions combining both above, e.g., real-time systems of airline tickets booking.	- Systems with complex mathematical algorithms or with other specialised and complex rules (e.g., expert, simulation, self-learning systems) - Systems processing continuous variables (audio, video) - For above-mentioned domains it is possible to modify the method so that it may be used locally.
ISO/IEC 29881 - FiSMA method	All software classes	None

Source: Author's own study based on [15, Part 6].

The FSMM standardized by the ISO/IEC provide sufficiently objective and reliable basis for BSS D&EP effort, budget and time frame estimating. Results of numerous surveys, including e.g., those carried out by the State Government of Victoria [24] and International Software Benchmarking Standards Group [17], indicate that BSS D&EP, in case of which the FSMM were used for effort planning, are characterised by relatively accurate estimations. Studies by the State Government of Victoria indicate that pricing of BSS on the basis of product size expressed in functionality units results in reducing the average budget overrun to less than 10% – comparing with current average budget overrun amounting to approx. 55% [2]. The ISBSG report confirms these results: in the situation where the methods based on product functional size are employed in making cost estimation, in 90% of cases the estimates differ from the actual costs not more than by 20%, and among these very cases 70% are accurate to within 10%. Also analysis of the results of 25 studies concerning the reliability of the most important BSS D&EP effort estimation methods, made by the author on the basis of the subject literature [25], revealed that currently the highest accuracy of effort estimations is delivered by the extrapolation methods based on software product size expressed in functionality units.

V. FUNCTIONAL ASSESSMENT OF BUSINESS SOFTWARE SYSTEMS DEVELOPMENT AND ENHANCEMENT PROJECTS

It is worth taking advantage of the capabilities offered by software FSM to the assessment of BSS D&EP from the perspective being fundamental to a client – that is from functional perspective.

A. The southernSCOPE and northernSCOPE methodologies

Here is why the FSM concept constitutes basis of the southernSCOPE [13] and northernSCOPE [26] methodologies, supporting the management of BSS D&EP functional scope, i.e., scope measured on the basis of functional size of their product. Fundamental assumptions of these methodologies read as follows (see also [27]):

- Price to be paid by client for software being accomplished within D&EP depends directly on the functional size of project product.
- Estimates are being derived throughout the project's life cycle.
- Structure of changes management promotes proper management of changes being introduced by client to the functional requirements.
- Person responsible for the scope management, the so-called scope manager, ascribed key role in this methodology, should work independently.

Practice shows that the discussed methodologies prove useful in the case of projects aimed at developing or enhancing BSS, regardless of whether or not they have internal or external character. As conditions of the effective use of these approaches are being met in their case; among these conditions are:

- Accomplishment of project within the planned and controlled budget is of key significance, if not a priority, to a client.
- There is an acceptance for the methods of product functional size measurement.
- Functional user requirements can be specified on the level of detailness suitable for the FSMM.
- There is a possibility to reduce the number of changes to the required product functionality appearing upon completion of the requirements specification phase.

Concurrently, with the above methodologies, the author of this paper proposed [27] and verified [28] her own model, designed for the functional assessment of BSS D&EP, named SoftFAM (Software projects Functional Assessment Model). Functional Assessment (FA) of project is understood by the author as its *ex ante* and *ex post* evaluation carried out on the basis of FSM concept. Key attributes of FA include: product functional size (*FS*), work effort, which needs to be spent on *FS* development/enhancement (*E*), and functional productivity (*P*) understood as the ratio of product functional size to the work effort on *FS* development/enhancement (*FS*/*E*) [29], or – being inversion of functional productivity - work effort necessary to achieve functionality unit ($E(u)=E/FS$) that determines work cost per *FS* unit (thus measured with regard to the product size unit, not to the work time unit).

B. Assumptions for SoftFAM

The SoftFAM may occur in the form of full model as well as in one of the simplified variants – thus it has a modular character. The following assumptions were explicitly included to the full variant of the model:

1. Functional assessment consists of at least three stages:
   1.1. Initial functional assessment (FA1). It may take place as soon as at the stage of initiating BSS D&EP thanks to the functional size early estimation rule, having been derived on the basis of benchmarking data [25][30] (the so-called calculations of Function Points Zero – FP0). Yet more accurate estimates are received at the analysis stage where the fundamental FUR are known – they are based on the calculations of FP1 (Function Points One), for which, according to the rules of FSM methods, estimation error up to ±30% is allowed. Estimation made at this very stage should be sufficient for initial planning of project attributes, making initial decision on investment, choosing execution variant as well as for choosing group of providers' offers. Further analytical works involve substantial means, which - according to the ISBSG report [31] - make up even up to approx. 27% of the effort spent during the entire project cycle and thus it is worthwhile to make use of the possibility to rationalize of these activities and decisions already at this very stage.
   1.2. Detailed functional assessment (FA2). For the second time estimation should be carried out when detailed FUR specification is already known, which is upon completion of the analytical stage. At this stage estimations are based on calculating FP2 (Function Points Two), in case of which – in accordance with the FSM methods rules – estimation error should not exceed ±10%. Thus, what should be done is a correction of the initially estimated required functional size and based on this – the required effort and functional productivity. This correction results not only from the fact of FUR changing since the moment of calculating FP1 but also from the change of the error range allowed for *FS* at this very stage and consequently – also for the attributes estimated on the basis of *FS*. Based on estimations being derived at this stage, another functional assessment of the previously selected group of providers' offers should be made so that as a result at most several potential product providers will be chosen following the criteria of such assessment. Selecting one of these providers may depend on other criteria as well – they should regard first of all fulfilling of client's non-functional

requirements. It is important that the required product functional size as well as the offered and approved work cost per functionality unit are reflected in provider's formal commitment to a client, which means formal *ex ante* pricing of the project product.

1.3. Final functional assessment (FA3). For the third time functional assessment should be made upon completion of development/enhancement activities in order to measure the actually delivered *FS*, which is meant to lead first of all to the *ex post* pricing of product on the basis of this size and the approved work cost per functionality unit as well as it is to be used to verify degree of FUR accomplishment by a provider, who thus gains possibility to enhance his software processes. Data obtained this way should be then stored by provider in the organizational benchmarking data repository, especially designed for this very purpose. This is meant for deriving and verifying dependencies being specific to given project organization but also for enhancing FSM methods and effort estimation models. At this stage calculations should take into account the fact that since the moment of making FP2 calculations FUR might have changed. Thus the value of all required attributes needs to be updated.

2. All *r*equired (*FSr*, *Er*, *Pr*), *o*ffered (*FSo*, *Eo*, *Po*) and *re*alised (*FSre*, *Ere*, *Pre*) attributes should be included to the relevant tolerance intervals, dependent on the functional assessment stage, which normalize the ranges of allowed values. The need of taking them into account results both from the limited possibilities to derive accurate estimates, particularly at the initial assessment stage, being caused first of all by the BSS D&EP execution conditions changing over time, as well as by analytical needs. Tolerance intervals should promote rational delineating of required and offered attributes values. They read as follows:

2.1. Product functional size – both required by a client (*FSr*) as well as offered (*FSo*) and realised (*FSre*) by a provider – must be within the range allowed for *FSr*, i.e., [*FS*min, *FS*max], where: *FS*min – stands for minimum while *FS*max – stands for maximum required functional size. Defining of *FS*max results from the fact that, as showed by the Standish Group studies, only about 20% of functions and features specified get ever used [2]. Thus delineating the maximum expected functional size reduces the risk of delivering needless functionality.

2.2. Work effort – both expected by a client (*Er*) as well as offered (*Eo*) and realised (*Ere*) by a provider – must be within the range allowed for *Er*, i.e., [*E*min, *E*max], where: *E*min – stands for minimum while *E*max – stands for maximum effort expected by a client. *E*min should not be lower than the effort enabling for delivering minimum required functional size (*FS*min).

2.3. Functional productivity – both required by a client (*Pr*) as well as offered (*Po*) and realised (*Pre*) by a provider – must be within the range allowed for *Pr*, i.e., [*P*min, *P*max], where: *P*min - stands for minimum while *P*max - stands for maximum productivity required by a client. Having *P*max defined is useful for rational provider offer selection, i.e., from the point of view of limiting the risk of choosing the offer where the productivity would be defined as overstated value. Since such situation would mean that in fact the effort per functionality unit is likely to be exceeded, which would entail the risk of delivering product having functional size lower than the allowed one as the provider would be probably trying not to go over the offered effort. In addition, delineating *P*max is conducive to the increased probability of delivering product of sufficient quality.

Fulfilling these conditions ensures:

- Rationality of client requirements with regard to the functional assessment attributes.
- Conformity of the potential providers offers with rational client requirements concerning functional assessment attributes.
- Conformity of the accomplished project with client requirements concerning functional assessment attributes.

The full variant of SoftFAM comprises at least two stages of estimation (FA1, FA2), within which the ranges of allowed values for functional attributes are being used. Due to the modular character of the presented model there is also the possibility to use its simplified variants, which may be considered for applying in practice keeping in mind, however, the increase of risk caused by such simplification. As indicated by the analysis in [27], level of satisfying client's analytical needs decreases with gradual resignation from, initially, one of the two stages of assessment, next from the intervals of allowed values for functional size, effort and functional productivity, and then with omitting both aspects of the FA. Assessment will be more detailed if a client resigns from the initial stage of estimation thus, however, increasing the risk of making non-rational investment decision due to the estimates being delayed in relation to the possibilities.

C. Verification of SoftFAM

The verification of the full variant of SoftFAM was based on the case study of a dedicated BSS being developed from scratch for the needs of Polish affiliated sales department of some international motor concern and presented widely in [28].

Results of the verification indicate that SoftFAM allows for *ex ante* and *ex post* assessment of BSS D&EP effectiveness, and it also supports *ex ante* and *ex post*

analysis of BSS D&EP economic efficiency. As these results prove that functional assessment allows rationalizing certain practical activities as well as business decisions made on the basis of its criteria. Among such activities are: specification of rational client requirements concerning key project attributes (product size, project work effort, cost and time), evaluation of potential providers offers, comparison of execution variants from the point of view of estimated work costs and the economic efficiency, indicating variant having highest potential efficiency, rational *ex ante* and *ex post* pricing of project product as well as enhancing prognosis concerning future projects by project provider. Among business decisions being supported by functional assessment should be mentioned: client's investment decision about going into the execution of project having expected attributes, selection of the offer being most adequate to his requirements concerning these attributes as well as selection of execution variant having highest economic efficiency.

Moreover, results of the verification also indicate that formal pricing of BSS D&EP product should base on the required size (*ex ante* pricing) and on the actually delivered size (*ex post* pricing) of this product expressed with the use of functionality units and on the work costs per unit being measured with regard to the product size unit – and not on the fixed price contracts nor time and material contracts, most often occurring in the project practice, not only in Poland [14, p. 250], which promote exceeding of the BSS D&EP execution costs.

Because of the above capabilities, the SoftFAM allows for reducing some of the negative phenomena commonly occurring in the Polish practice of such projects execution, showing negative influence on their effectiveness and also on their real efficiency, namely:

- Deliberate lowering of BSS delivery costs by providers in order to win contract for product development (the so-called "price-to-win" technique for product pricing) – thanks to *ex ante* and *ex post* product pricing based on the required and actually delivered product functional size and work cost per functionality unit having been mutually and formally agreed at the stage of provider selection.
- Clients increasing the required functionality during the project lifecycle without relevant reflecting of this change's consequences in the execution costs – as a result of monitoring each change in product functional size and ability to determine this change's influence on total work costs on the basis of the formally agreed work cost per functionality unit.
- Provider in reality delivering product having functionality lower than the required one within the fixed price contracts – client is not obligated to pay for the functionality, which had not been delivered as the *ex post* product pricing is based on its actually delivered functional size.
- Provider delivering functionality (many a time also being lower than the required one) at costs being higher than those expected, which usually takes place in the case of time and material contracts – client does not settle the payment on the basis of project duration but on the basis of actually delivered product functional size and formally agreed work cost per functionality unit.

This is possible thanks to the following rules being used in the full variant of SoftFAM:

- Adopting the allowed tolerance intervals for required, offered and realised FA attributes.
- When choosing offers for project execution, preferring the highest allowed productivity (the lowest allowed effort per functionality unit) instead of the cheapest offers.
- Taking into account the influence of changes in FUR being made during the project lifecycle on product functional size, work effort and functional productivity.
- *Ex ante* and *ex post* pricing of product based on the required and actually delivered product functional size as well as mutually agreed work cost per functionality unit.

Verification of the full SoftFAM indicates that it promotes fundamental factors of the effective execution of BSS D&EP [2] – as it contributes to getting client involved in the project and to the proper management of project scope, as well as to achieving most of the functional measurement goals mentioned in the ISO/IEC 14143 norm, especially in the area of project management [15, Part 6].

Advantage of the full version of SoftFAM over southernSCOPE and northernSCOPE methodologies results from the fact of the model adopting two significant assumptions, not being explicitly specified in these methodologies, namely (see also [27]):

- Need to apply upper bounds of the allowed tolerance intervals for required, offered and realised functional size and functional productivity and lower bounds for work effort.
- Need to employ at least two stages of estimation: first one for proper assessment of the investment decision rationality while second stage – in order to choose suitable software product provider.

Therefore, comparing to these methodologies, using full SoftFAM reduces the risk of choosing inappropriate provider as well as the risk of lowered *ex ante* and overstated *ex post* product pricing, and consequently, it reduces the chance of failing to deliver required functionality and/or to deliver product of insufficient quality. On the other hand, modular character of SoftFAM enables for choosing its variant being most suitable to a given situation – it may be a version based on the simplest criteria, closest to the southernSCOPE and northernSCOPE methodologies.

VI. USAGE OF FUNCTIONAL SIZE MEASUREMENT METHODS BY POLISH BUSINESS SOFTWARE SYSTEMS PROVIDERS

A necessary condition for taking advantage of BSS D&EP functional assessment is to employ software FSM methods. Meanwhile, the author's studies, whose results were widely presented in [4], indicate that the level of using

these methods among Polish BSS providers, although growing, still leaves a lot to be desired.

Surveys that aimed at analysing the level of using the software FSMM by the Polish BSS providers as well as the reasons behind this status quo, were conducted against a background of author's own research concerning the usage of BSS D&EP effort estimation methods. The use of both types of methods was examined in two cycles: at the turn of the year 2005/2006, being the time of economic prosperity, and next at the turn of the year 2008/2009, that is in the initial stage of crisis and increased investment uncertainty associated with it (in order to observe changes, the author originally intended the research to be repeated after 5 years, however radical change in the economic situation worldwide and in Poland persuaded her to undertake it 2 years earlier).

Both research cycles were completed using the method of diagnostic survey: the first cycle analysed responses given in 44 questionnaires (52 questionnaires were sent out) while the second cycle – responses given in 53 questionnaires (62 questionnaires were sent out). Questionnaires were distributed among various Polish dedicated BSS providers, both internal (IT departments in organizations) as well as external (for the most part from SME sector), providing systems for the needs of financial institutions (banks, insurance) departments, trading companies and public administration institutions. In both cycles the overwhelming majority of responses were answered by IT managers or project managers. Each questionnaire included about 30 questions validated by experts; most questions were of open or semi-open character and were divided into two main groups: concerning the usage of the effort estimation methods (answered by all respondents) and concerning the usage of the FSMM (answered only by the respondents familiar with FSMM). It should be stressed that the research was limited only to organizations dealing with D&EP, whose products are dedicated BSS – thus analysis included neither software maintenance, support and integration projects, software package acquisition and implementation projects, nor other software products types.

In the context of the subject matter analysed in this paper fundamental conclusions from these surveys read as follows:

- Considerable part of the respondents declares they do not commonly employ any of the methodology-based approaches to the BSS D&EP effort estimation, in most cases pointing to the "price-to-win" technique as the preferred estimation approach (not methodology-based) when providing software systems for government institutions (because of legal regulations). However, the level of using the BSS D&EP effort estimation methods has increased over the analysed time (from 45% to 53% of the surveyed providers).
- In both research cycles the respondents declared rather widespread usage of at least one of the effort estimation methods, mostly pointing to the expert methods (first cycle: 36%, second cycle: 43% of all respondents), which are burdened with high risk (tests show that the ratio of the effort estimates, being calculated by different experts for the same project may be 1:6 or even 1:12 at the worst [32]).
- FSM methods still place at the penultimate position among five analysed methods used for BSS D&EP effort estimation by the surveyed providers, however the level of using them has increased in the second research cycle (from 20% to 26% of all respondents).
- In both research cycles relatively low popularity of the FSMM results mostly from insufficient familiarity with such methods, but the FSMM awareness has increased over the analysed time (from 27% to 34% of all respondents).
- Percentage of the respondents using FSM methods versus those familiar with them has increased slightly too (from 75% to 78%), which means that the overwhelming majority of those familiar with the FSMM are also employing them.
- In both research cycles as the main purpose of using the FSM methods was considered product size estimation in order to effectively estimate the effort, costs and time frame for the initiated project.
- In both research cycles as the main advantages of the FSM methods were considered the methods objectivity and high usefulness, including most of all possibility to employ them at initial project stages at sufficient accuracy level of estimates, which helps increase the effectiveness of delivering the required functionality on time and within the planned budget. Disadvantages of the FSM methods include first of all high level of difficulty in using them.

As indicated by the above, in the case of all respondents the main reason for relatively low popularity of the FSM methods is that none of the BSS D&EP effort estimation methods is used commonly as well as insufficient familiarity with these methods, whereas among respondents using estimation methods – insufficient awareness of FSMM and at the same time familiarity with other methodology-based approaches. Among providers declaring familiarity with the FSM methods the main reason why they quitted using them is their high difficulty level.

Fundamental purposes for using the FSM methods indicated by the surveyed Polish dedicated BSS providers are presented in Table 3, where they are related to the purposes for using FSM described in the ISO/IEC 14143 norm. Data presented in Table 3 indicate that (see also [4]):

- In both research cycles higher importance is assigned to the purposes of Project Management group.
- Fundamental purpose of using FSMM indicated in both research cycles is product size estimation in order to effectively estimate the effort, costs and time frame for the initiated project, which is the purpose belonging to the Project Management group.
- Among purposes belonging to the Performance Management group, productivity management was indicated as the most important one in both research cycles.

TABLE III. BASIC PURPOSES FOR USING THE FSM METHODS INDICATED BY THE SURVEYED POLISH DEDICATED BSS PROVIDERS

Purpose indicated by Polish BSS providers	2006 (%)	2009 (%)	ISO/IEC 14143 purpose	
Estimation of product size and, based on this, estimation of the effort, costs and time frame for the project being initiated – in order to design own offer as well as for the commissioned applications	100%	100%	Project resource forecasting	Project Management
Supporting decisions about rationality of initiating the projects and way of completing projects (e.g., using own resources or by outsourcing)	56%	64%		
Monitoring progress, costs and time in the project execution	67%	64%	Tracking the progress of a project	
Managing the changes in the required product size and their influence on project work effort	44%	36%	Managing scope change	
Determining degree to which the Commercial-Off-The-Shelf meets functional user requirements	0%	7%	Package functionality fit	
Comparing attributes of the finished project with other projects	44%	50%	Post-mortem analysis	
Managing software development, enhancement or maintenance productivity	78%	86%	Productivity management	Performance Management
Managing software reliability	44%	50%	Quality management	
Managing organization's maturity	0%	7%	Organizational maturity and process capability	
Measuring existing applications in order to determine their value to estimate costs of its potential replacement, reengineering, or outsourcing	56%	64%	Accounting for an organization's software asset	
Making prognosis on the budget necessary to maintain software	33%	29%	Budgeting for maintenance	
Managing the product size and project scope in the client-provider relations	67%	78%	Contract management	
Valuation of applications being executed by other companies	56%	57%		
Determining degree to which the supplied dedicated product meets functional user requirements	0%	14%		

Source: Author's own study with the use of [15, Part 6].

- In 2009, three new items appeared on the list of purposes for using FSMM, namely: managing organization maturity and determining degree to which the supplied dedicated product *or* the COTS meets functional user requirements – in the first cycle they were indicated by none of the surveyed Polish dedicated BSS providers.

The FSM methods stayed practically unknown in Poland until the recession in IT branch that took place in the first years of the 21st century. Although the level of using these methods can be hardly considered high, increase in their popularity, however, may be possibly explained by the four main factors, namely:

- Increasing care about financial means in the times after recession mentioned above (including current crisis where it appears even somewhat stronger).
- Growing competition on the market and increasing market globalization level.
- Growing awareness of clients therefore greater requirements concerning providing justification for the project costs and completion time offered by providers.
- Standardization of the FSM concept and its several methods by the ISO/IEC.

It is hard to compare conclusions coming from the above analysis with the results of other studies carried out worldwide in this area, as the author heard no about studies having similar goals. Yet the fundamental conclusion brought by these surveys agrees with the general conclusion drawn by the Software Engineering Institute (SEI) on the basis of the research attempted to answer the question about today's approach to the measurement of software processes and products: "From the perspective of SEI's Software Engineering Measurement and Analysis (SEMA) Group, there is still a significant gap between the current and desired state of measurement practice. (…) Generally speaking, based on the results of this survey, we believe that there is still much that needs to be done so that organizations use measurement effectively to improve their processes, products, and services." [33].

The research will be continued to keep observing the changes while the research area will be extended as much as possible to other Polish dedicated BSS providers and other economic BSS D&EP aspects.

VII. CONCLUSION AND FUTURE WORK

Summing up it should be stated that the importance of suitable BSS size measurement being significant from the economic point of view results first of all from the necessity to:

1) Increase effectiveness of BSS D&EP execution and reduce losses caused by their low effectiveness. Accurate *ex ante* assessment of project product size, cost and time increases the chance to reach its goal, i.e., on-time delivery of BSS being consistent with client's business requirements without budget overrun. Since the more accurate estimation the lower the risk to go beyond estimates in reality. What's more, such assessment enables to get information about resources that are necessary to deliver product having required functions and features – and it should allow for quitting projects, for which the chance of execution with the resources available proves low, or for correcting resources designed for the projects so that they are closest to the

estimated values. Down to the more accurate investment decisions made on the basis of measurable, objective and reliable criteria it is possible to reduce losses caused not only by abandoned projects and by large scale of overrunning the time and costs of their execution but also resulting from business opportunities lost by clients as a result of delivering products not meeting their requirements.

2) Rational ex ante and ex post pricing of BSS D&EP product. In the Polish practice of the BSS D&EP execution there are two types of client-provider contracts that definitely dominate at the moment, they are: fixed price contract and time and material contract. In the first case price of the project product is calculated on the basis of the assumed fixed costs, which were agreed following the requirements specification. In contracts of another type calculation of the product price is based on the agreed rate for work hour being spent by product provider. It means that work cost per unit is measured not with regard to the unit of product size but with regard to the unit of work time, and therefore this is work time – instead of required or actually delivered product size – that determines the total work costs. Project execution with *ex post* pricing of actually delivered product is still rare, at least in Poland, where we deal with low (however growing) level of the so-called "measurement culture" in software engineering, especially from the functional point of view (see Section 6). Both these approaches to the BSS pricing promote overrunning of budget designed for delivering of product that would meet client's requirements. In case of client-provider contracts based on hourly work rate the provider could extend the time of product execution. Also, there is no guarantee that even extending this time excessively and thus leading to the uncontrolled increase in costs the provider would deliver product of required functionality. In case of fixed price contracts, apart from likely situation where the actually delivered product size may be smaller than the required one, there is also another problem that arises: providers manifest strong resistance to any extension of requirements, being so characteristic of BSS D&EP due to the changeability of business environment. Thus the contracts of this type may prevent cost overrun yet on the other hand they do not guarantee delivering of product having required functions and features at this very cost. Therefore *ex ante* and *ex post* pricing of the BSS, being developed or enhanced, should be based on its size: required (estimated) in the case of *ex ante* pricing and actually delivered (measured) in the case of *ex post* pricing. Consequently, work costs per unit should be related to the product size unit and not to the work time unit. This is what makes pricing have objective and reliable character, as client will get possibility to plan the cost of project execution depending on the outcome this project is expected to bring and, as a consequence of its execution, will pay for the actually delivered size of product and not for his requirements, which provider failed to fulfil (in case of fixed price contracts) or for the provider's extra work time (in case of time and material contracts). It requires adequate measure of software size to be implemented, which may be acquired on the basis of the software functional size measurement concept, having been recently normalised by the ISO/IEC.

3) Proper control over the BSS D&EP execution. Measuring product size and project attributes during project execution helps perceive discrepancies between the reality and the plan, respond to potential threats on a current basis, prevent risk factors and monitor the areas of critical significance.

4) Collecting historical data for BSS D&EP estimation purposes. Measurement of the accomplished BSS D&EP attributes allows for deriving dependencies indispensable for making accurate estimation of similar projects in the future thus leading to the enhancement of estimation models that are based on such dependencies.

5) Improvement of BSS D&EP products and processes. Capability to measure software quality (e.g., reliability, what requires knowing the product size) allows to specify client's quality requirements with the use of quantitative criteria, to carry out measurable assessment of product quality during project lifecycle, thus making it possible to verify whether its level is satisfactory, what may result in undertaking improvement activities, as well as to make quality assessment of the final product. On the other hand, SPA/SPI (Software Process Assessment/Software Process Improvement) models (e.g., CMMI - Capability Maturity Model Integration) are based on the assumption that better software product is achieved by means of the improved software processes [34], whose quality too requires to be assessed. In these models higher and higher importance is attached to the software products and processes measurement.

From the point of view of software organizations the measurement of products and processes should be a standard practice: estimating and measuring product size, process effort, cost and time enable for more effective business activity. Estimating and measurement prove being very important also from the point of view of these organizations' clients, who should be given grounds for making rational investment decision and consequently for choosing variant promoting minimisation of costs at the assumed level of effects (required product size), possibly maximisation of effects (achievable product size) at the assumed costs level (if unexceedable costs were determined *a priori*). Moreover, experience in the Polish market (yet not only in this one) indicates that in the practice of BSS D&EP we still cannot speak about the balance of power between a provider and client. The former often dictates conditions of cooperation, many a time making use of client ignorance, especially with regard to the BSS pricing, imposing – if only client allows for it – contract conditions being favourable for himself.

Change of this situation is possible owing to employing suitable approach to the BSS size measurement, that is functional approach, and thanks to taking advantage of the capabilities offered by FSM concept and methods for the BSS D&EP assessment from the perspective being of key significance to a client. Therefore the author made an attempt to develop SoftFAM – the model of BSS D&EP functional assessment that would allow for evaluating the

effectiveness of their execution, both *ex ante* as well as *ex post*, and for supporting *ex ante* and *ex post* analysis of BSS D&EP economic efficiency. The SoftFAM verification results prove that such model allows rationalizing certain practical activities and business decisions made on the basis of its criteria, as well as it allows for reducing some of the negative phenomena commonly occurring in the practice of such projects execution, not only in Poland.

REFERENCES

[1] B. Czarnacka-Chrobot, "The Economic Importance of Business Software Systems Size Measurement", Proceedings of the 5th International Multi-Conference on Computing in the Global Information Technology (ICCGI 2010), 20-25 September 2010, Valencia, Spain, M. Garcia, J-D. Mathias, Eds., IEEE Computer Society Conference Publishing Services, Los Alamitos, California-Washington-Tokyo, 2010, pp. 293-299.

[2] Standish Group, "CHAOS summary 2009", West Yarmouth, Massachusetts, 2009, pp. 1-4.

[3] David Consulting Group, "Project estimating", DCG Corporate Office, Paoli, 2007: http://www.davidconsultinggroup.com/training /estimation.aspx (20.05.2011).

[4] B. Czarnacka-Chrobot, "Analysis of the functional size measurement methods usage by Polish business software systems providers", in Software Process and Product Measurement, A. Abran, R. Braungarten, R. Dumke, J. Cuadrado-Gallego, J. Brunekreef, Eds., Proc. of the 3rd International Conference IWSM/Mensura 2009, Lecture Notes in Computer Science, vol. 5891, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 17–34.

[5] T. C. Jones, Patterns of software systems failure and success, International Thompson Computer Press, Boston, MA, 1995.

[6] PCG, "2008 ERP report, topline results", Panorama Consulting Group, Denver, 2008, pp. 1-2.

[7] B. Czarnacka-Chrobot, "Standardization of Software Size Measurement", in Internet – Technical Development and Applications, E. Tkacz and A. Kapczynski, Eds., Advances in Intelligent and Soft Computing, vol. 64, Springer-Verlag, Berlin-Heidelberg, 2009, pp. 149-156.

[8] J. Johnson, "CHAOS rising", Proc. of 2nd Polish Conference on Information Systems Quality, Standish Group-Computerworld, 2005, pp. 1-52.

[9] Standish Group, "CHAOS summary 2008", West Yarmouth, Massachusetts, 2008, pp. 1-4.

[10] Economist Intelligence Unit, "Global survey reveals late IT projects linked to lower profits, poor business outcomes", Palo Alto, California, 2007: http://www.hp.com/hpinfo/newsroom/press/2007/ 070605xa.html (20.05.2011).

[11] Standish Group, "The CHAOS report (1994)", West Yarmouth, Massachusetts, 1995, pp. 1-9.

[12] T. C. Jones, "Software project management in the twenty-first century", Software Productivity Research, Burlington, 1999, pp. 1-19.

[13] State Government of Victoria, "southernSCOPE, reference manual", Version 1, Government of Victoria, Melbourne, Australia, 2000, pp. 1-22.

[14] M. A. Parthasarathy, Practical software estimation: function point methods for insourced and outsourced projects, Addison Wesley Professional, 2007.

[15] ISO/IEC 14143 Information Technology – Software measurement – Functional size measurement – Part 1-6, ISO, Geneva, 2007.

[16] Gartner Group, "Function points can help measure application size", Research Notes SPA-18-0878, 2002.

[17] ISBSG, "The ISBSG report: software project estimates – how accurate are they?", International Benchmarking Standards Group, Hawthorn VIC, Australia, 2005, pp. 1-7.

[18] ISO/IEC 20926 Software and systems engineering - Software measurement - IFPUG functional size measurement method 2009, ISO, Geneva, 2009.

[19] ISO/IEC 20968 Software engineering – Mk II Function Point Analysis - Counting practices manual, ISO, Geneva, 2002.

[20] ISO/IEC 24570 Software engineering – NESMA functional size measurement method version 2.1 - Definitions and counting guidelines for the application of Function Point Analysis, ISO, Geneva, 2005.

[21] ISO/IEC 19761 Software engineering – COSMIC-FFP – A functional size measurement method, ISO, Geneva, 2003.

[22] ISO/IEC 29881 Information Technology – Software and systems engineering – FiSMA 1.1 functional size measurement method, ISO, Geneva, 2008.

[23] B. Czarnacka-Chrobot, "The ISO/IEC Standards for the Software Processes and Products Measurement", in New Trends in Software Methodologies, Tools and Techniques, H. Fujita and V. Marik, Eds., Proc. of the 8th International Conference SOMET'2009, Frontiers in Artificial Intelligence and Applications, vol. 199, IOS Press, Amsterdam-Berlin-Tokyo-Washington, 2009, pp. 187-200.

[24] P. R. Hill, "Some practical uses of the ISBSG history data to improve project management", International Software Benchmarking Standards Group, Hawthorn VIC, Australia, 2007, pp. 26-30.

[25] B. Czarnacka-Chrobot, "The role of benchmarking data in the software development and enhancement projects effort planning", in New Trends in Software Methodologies, Tools and Techniques, H. Fujita and V. Marik, Eds., Proc. of the 8th International Conference SOMET'2009, Frontiers in Artificial Intelligence and Applications, vol. 199, IOS Press, Amsterdam-Berlin-Tokyo-Washington, 2009, pp. 106-127.

[26] Finnish Software Metrics Association, "nothernSCOPE – customer-driven scope control for ICT projects", FiSMA, March 2007.

[27] B. Czarnacka-Chrobot, "Methodologies supporting the management of business software systems development and enhancement projects functional scope", Proc. of the 9th International Conference on Software Engineering Research and Practice (SERP'10), The 2010 World Congress in Computer Science, Computer Engineering & Applied Computing (WORLDCOMP'10), Hamid R. Arabnia, Hassan Reza, Leonidas Deligiannidis, Eds., Vol. II, CSREA Press, Las Vegas, Nevada, USA, July 2010, pp. 566-572.

[28] B. Czarnacka-Chrobot, "Rational pricing of business software systems on the basis of functional size measurement: a case study from Poland", Proc. of the 7th Software Measurement European Forum (SMEF) Conference, T. Dekkers, Ed., Libreria Clup, Rome, Italy, June 2010, pp. 43-58.

[29] L. Buglione, "Some thoughts on Productivity in ICT Projects, version 1.2", WP-2008-02, White Paper, July 25, 2008.

[30] "Practical Project Estimation (2nd edition): A Toolkit for Estimating Software Development Effort and Duration", P. R. Hill, Ed., ISBSG, Hawthorn, VIC, 2005

[31] ISBSG, "The ISBSG Special Analysis Report: Planning Projects – Project Phase Ratios", International Software Benchmarking Standards Group, Hawthorn, VIC, Australia, 2007, pp. 1-4.

[32] International Software Benchmarking Standards Group: http://www.isbsg.org/Isbsg.Nsf/weben/Functional%20Sizing %20Methods (20.05.2011).

[33] M. Kasunic, "The state of software measurement practice: results of 2006 survey", Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2006, pp. 1-67.

[34] CMMI Product Team, "CMMI for Development", Version 1.2, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, 2006, pp. 1-573.

www.iariajournals.org

International Journal On Advances in Intelligent Systems
ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2679

International Journal On Advances in Internet Technology
ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
issn: 1942-2652

International Journal On Advances in Life Sciences
eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
issn: 1942-2660

International Journal On Advances in Networks and Services
ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
issn: 1942-2644

International Journal On Advances in Security
ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
issn: 1942-2636

International Journal On Advances in Software
ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS, CLOUD COMPUTING, COMPUTATION TOOLS
issn: 1942-2628

International Journal On Advances in Systems and Measurements
ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
issn: 1942-261x

International Journal On Advances in Telecommunications
AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
issn: 1942-2601