International Journal on

Advances in Internet Technology

2015 vol. 8 nr. 1&2

The *International Journal on Advances in Internet Technology* is published by IARIA.
ISSN: 1942-2652
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Internet Technology
Volume 8, Number 1 & 2, 2015

Jorge Sá Silva, University of Coimbra, Portugal
Sébastien Salva, University of Auvergne, France
Ahmad Tajuddin Samsudin, Telekom Malaysia Research & Development, Malaysia
Josemaria Malgosa Sanahuja, Polytechnic University of Cartagena, Spain
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías / University of Castilla-La Mancha, Spain
Paul Sant, University of Bedfordshire, UK
Brahmananda Sapkota, University of Twente, The Netherlands
Alberto Schaeffer-Filho, Lancaster University, UK
Peter Schartner, Klagenfurt University, System Security Group, Austria
Rainer Schmidt, Aalen University, Germany
Thomas C. Schmidt, HAW Hamburg, Germany
Didier Sebastien, University of Reunion Island, France
Zary Segall, Chair Professor, Royal Institute of Technology, Sweden
Dimitrios Serpanos, University of Patras and ISI/RC ATHENA, Greece
Jawwad A. Shamsi, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan
Michael Sheng, The University of Adelaide, Australia
Kazuhiko Shibuya, The Institute of Statistical Mathematics, Japan
Roman Y. Shtykh, Rakuten, Inc., Japan
Patrick Siarry, Université Paris 12 (LiSSi), France
Jose-Luis Sierra-Rodriguez, Complutense University of Madrid, Spain
Simone Silvestri, Sapienza University of Rome, Italy
Åsa Smedberg, Stockholm University, Sweden
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Radosveta Sokullu, Ege University, Turkey
José Soler, Technical University of Denmark, Denmark
Victor J. Sosa-Sosa, CINVESTAV-Tamaulipas, Mexico
Dora Souliou, National Technical University of Athens, Greece
João Paulo Sousa, Instituto Politécnico de Bragança, Portugal
Kostas Stamos, Computer Technology Institute & Press "Diophantus" / Technological Educational Institute of Patras, Greece
Vladimir Stantchev, SRH University Berlin, Germany
Tim Strayer, Raytheon BBN Technologies, USA
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Tae-Eung Sung, Korea Institute of Science and Technology Information (KISTI), Korea
Sayed Gholam Hassan Tabatabaei, Isfahan University of Technology, Iran
Yutaka Takahashi, Kyoto University, Japan
Yoshiaki Taniguchi, Kindai University, Japan
Nazif Cihan Tas, Siemens Corporation, Corporate Research and Technology, USA
Alessandro Testa, University of Naples "Federico II" / Institute of High Performance Computing and Networking (ICAR) of National Research Council (CNR), Italy
Stephanie Teufel, University of Fribourg, Switzerland
Parimala Thulasiraman, University of Manitoba, Canada
Pierre Tiako, Langston University, USA
Ioan Toma, STI Innsbruck/University Innsbruck, Austria
Orazio Tomarchio, Universita' di Catania, Italy

Kurt Tutschku, University Blekinge Institute of Technology, Karlskrona, Sweden
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
Massimo Villari, University of Messina, Italy
Krzysztof Walkowiak, Wroclaw University of Technology, Poland
MingXue Wang, Ericsson Ireland Research Lab, Ireland
Wenjing Wang, Blue Coat Systems, Inc., USA
Zhi-Hui Wang, School of Softeware, Dalian University of Technology, China
Matthias Wieland, Universität Stuttgart, Institute of Architecture of Application Systems (IAAS),Germany
Bernd E. Wolfinger, University of Hamburg, Germany
Chai Kiat Yeo, Nanyang Technological University, Singapore
Mark Yampolskiy, Vanderbilt University, USA
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

International Journal on Advances in Internet Technology
Volume 8, Numbers 1 & 2, 2015

CONTENTS

Services to Support Use and Development of Multilingual Speech Input

António Teixeira, Pedro Francisco, Nuno Almeida, Carlos Pereira, and Samuel Silva
Department of Electronics, Telecommunications & Informatics / IEETA
University of Aveiro
Aveiro, Portugal
{ajst, goucha, nunoalmeida, cepereira, sss}@ua.pt

***Abstract*—The use of speech for human-machine interaction benefits from being our most natural way of communication. Profiting from a widespread use of devices providing audio input, applications increasingly support speech recognition and understanding, but several challenges are still to be addressed. A common solution to support spoken language understanding uses semantic grammars. Also, for speech recognizers, grammars are a common way of configuring what can be recognized. The definition of these grammars, to support a semantically rich spoken interaction, involves a large amount of resources and linguistic knowledge and, therefore, systems tend to support a single language or multiple languages in a very narrow semantic scope. To address this issue, we propose to use an existing grammar to generate grammars for other languages by using automatic translation and taking into consideration aspects such as word realignment and grammar expansion, based on multiple possible translations. This method fosters effortless creation of a semantically rich grammar, for the target language, which can then be revised. The proposed method has already been successfully used to enable multilingual speech interaction for AALFred, a personal life assistant, covering English, French, Hungarian, Polish, and Portuguese.**

Keywords - Services; speech; semantic grammars; multilingual; multimodal interaction; speech recognition, translation.

I. INTRODUCTION

Advances in technology have brought mobile devices to our everyday life. With the growing number of features provided by devices such as smartphones or tablets, it is of paramount importance to devise natural ways of interacting with them that help to deal with their increasing complexity. Natural interaction is, therefore, an important goal, striving to integrate devices with our daily life by using gestures, context awareness or speech [1].

The importance of natural interaction is also boosted by the needs of various user groups, such as the elderly, who might present some kind of limitation at physical (e.g., limited dexterity) or cognitive (e.g., memory) level and lack the technological skills to deal with devices that can play an important role in improving their daily life [2].

The increased mobility and the multitude of devices that can be used impose important challenges to interaction design. Nevertheless, the "always connected" nature of most of these devices, in a wide range of environments (e.g., home, work, and street), offers the possibility of using resources located remotely, including computational power, storage or on-the-fly updates to currently running applications to serve a new context

Speech and natural language remain our most natural form of interaction [3][4] and a number of recent applications use speech as part of a multimodal system [5] in combination with other modalities. Nevertheless, despite its potential, the inclusion of input and output modalities based on speech poses problems at different levels. On a higher level, speech modalities involve several complex modules that need to work together and ensure speech recognition and speech synthesis. Tailoring these modules to different applications is a tiresome task and we have recently proposed a generic, service-based, modality component [6] that can work decoupled from the application, thus providing easier deployment of speech modalities. Another important issue concerning speech is its inclusion in applications targeting multiple languages. Therefore, our generic modality component also aims at being able to internally handle several languages. Since configuration information for speech recognizers or extraction of semantic information is created based on the context of one application, it is necessary to explicitly define the grammars of each application in each language that the application aims to support.

One of the demanding tasks on using the speech modality, when several languages are involved, is to help developers and user interaction designers in the derivation of the grammars for each language. Therefore, in this context of multi-language support, our main goals for the generic speech modality include:

- Streamlining of internationalization support;
- Reduction of variance among grammars, contributing for easier update and maintenance;
- Customization of any of the different grammars, if needed;
- Additionally to manual editing, allow automatic expansion of the recognized sentences and word corpora using existing services.

To approach these goals and in the context of a multimodal personal assistant, AALFred [7], part of project PaeLife, we proposed [1] the use of an existing grammar to generate grammars for other languages by using automatic translation and developed a first instantiation of a service which explores this idea to provide initial versions for the grammars in the different languages based on the definition of the semantic grammar (in English). The service receives a grammar, translates it and supports the needs of the speech modality.

The option for a service based solution follows a strong recent trend. Many mobile applications adopt cloud services, extending the capabilities of the device [8] regarding storage and processing capabilities. Also, the use of services to support the functionalities in speech input has been adopted in several mobile architectures, such as the mentioned mTalk [9], SIRI [10], and Cortana [11]. None, to the best of our knowledge, explored the use of automatic translation of grammars to support multilingual speech input.

The remainder of this document is organized as follows: Section II presents some background information regarding the application scenario in speech technologies, multimodal architectures, and support for spoken interaction in such architectures; Section III describes the main aspects of the proposed service regarding its architecture and main components; Section IV discusses prototype implementation; Section V provides some application examples; finally, Section VI presents some conclusions and ideas for future work.

II. Background and Related Work

To help understand the proposed service and how it relates to recent work in the areas of multimodal interaction and spoken language interaction, some background information and related work are presented. Also, to start this section, we briefly present the applications scenario chosen for the proof-of-concept is made.

A. *The application Scenario: The AAL PaeLife project*

The PaeLife project [12], chosen as the application scenario, is aimed at keeping the European elderly active and socially integrated. The project developed AALFred [7], [13], a multimodal personal life assistant (PLA), offering the elderly a wide set of services from unified messaging (e.g., email and twitter) to relevant feeds (e.g., the latest news and weather information). The platform of the PLA comprises a personal computer connected to a TV-like big screen and a portable device, a tablet. One of the key interaction modalities of the PLA is speech; speech input and output is available in five European languages: French, Hungarian, Polish, English, and Portuguese.

B. *Speech and interaction*

Speech and natural language remain our most natural form of interaction [3][4] and a number of recent applications use speech as part of a multimodal system [5] in combination with other modalities. Nevertheless, despite its potential, the inclusion of input and output modalities based on speech poses problems at different levels. On a higher level, speech modalities involve several complex modules that need to work together and ensure speech recognition and speech synthesis. Tailoring these modules to different applications is a tiresome task and we have recently proposed a generic, service-based, modality component [6] that can work decoupled from the application, thus providing easier deployment of speech modalities. Another important issue concerning speech is its inclusion in applications targeting multiple languages. Therefore, our generic modality component also aims at being able to internally handle several languages.

Several well-known applications use speech. A representative example is mTalk [9], a multimodal browser developed by AT&T, to support the development of multimodal interfaces for mobile applications. Siri [10] and Google Voice Search [14] are other examples of speech enabled applications.

Automatic Speech Recognition (ASR) takes as input the speech signal and produces a sequence of words. Speech recognition engines are typically based in Hidden Markov Models [15], which provide a statistical model to represent the acoustic model for the utterances. In addition to the acoustic model, a language model or a grammar is also needed to define the language. Language models, such as the ones defined by the ARPA format, are statistical n-gram [16] models that describe the probability of word appearance based on its history. Grammars can be defined as a set of rules and word patterns which provide the speech recognition engine with the sentences that are expected. The Java Speech Grammar Format (JSGF) [17] and GRXML [18] are examples of grammar formats.

Although grammars are more limited in the amount of sentences that will be recognized, they are capable of being more specific to each particular context of use, which often translates to a more accurate recognition.

These models and grammar are language dependent and, therefore, require language specific training. Usually, acoustic models and language models are trained generically to support a broad part of the language. They only need to be trained once for each language.

The next phase after speech recognition is Spoken Language Understanding (SLU), having as goal to extract semantic information from the sequence of words produced by the speech recognizer. Even in a command and control scenario for speech interaction it is very useful to associate a semantic meaning such as “direction left” to the sequence of words “please turn to the left”. Several different types of SLU [19][20] have been proposed, which can be divided into major groups: knowledge-based and data-driven approaches. Knowledge-based solutions include semantically enhanced syntactic grammars, and semantic grammars, whereas data-driven approaches explore both generative models and conditional (nongenerative) models [19]. Despite the potential of data-driven approaches, it is common to use semantic grammars (a knowledge-based approach) in many applications. These approaches represent the semantic space by a set of templates represented by semantic frames. Each frame contains typed components called “slots” (or frame elements). The type of the slot specifies what kind of fillers it expects. The goal of this frame-based SLU is to choose the frame that best matches the sequence of words coming from the speech recognizer and extract the values for the slots [19].

C. *The Multimodal Architecture and Speech Modality*

The support for multilingual spoken language interaction must be part of the multimodal architecture supporting interaction by voice. We have been working on the application of such architecture - directly related to the recent work of the W3C on a distributed architecture for multimodal interaction [21] - to mobile and AAL applications as described in [5][22].

One of the major advantages of this architecture is the decoupled nature of the interaction modalities, which enables the development of the application core without an explicit consideration of which modalities might be involved. What strictly matters is the semantic content resulting from the interactions, i.e., it does not matter if the user says "Go left" or presses the left arrow key in a keyboard. The application receives 'LEFT' for both. In this context, in a multilingual scenario, the application core stays the same and the support for new languages is added to the speech modality, which makes it easier to improve, in parallel with application development, or at a later stage, profiting from more advanced processing methods. The decoupled nature of this generic speech modality also enables its use in any application adopting the multimodal architecture.

III. SYSTEM OVERVIEW

The system's main objective is to be able to automatically generate a derived grammar in other target languages. That is achieved by preserving as much of the main grammar structure as possible, generating coherent phrases in the target language, and having in consideration the process of word reordering.

.

The system is dual in functionality. It supports both development and use in real interaction contexts.

In the development stage, developers use the system to make semantic grammars available and to produce the translated versions of such grammars. At this stage the service can also be used remotely to check and make corrections to the grammars. This can be done by native speakers or, if available, language specialists.

In interaction contexts, the system is in charge of the SLU, making use of the grammars sent to the service at development stage. It receives the output of speech recognition and returns the semantic information extracted. The service also returns, on request, to the speech modality, the necessary information on words and sentences needed to configure the speech recognition engine.

A. Architectural Definitions

The architecture, in Fig. 1, is composed of four main components: the speech modality, the core service, the access APIs, and the external resources (both parser and translator services). Further details about each component are provided in what follows.

1) Speech Modality

The speech modality is aligned with modalities in the multimodal architecture. It is a generic modality [6] able to provide speech interaction to applications that adopt the Multimodal Framework [23].

The modality allows the recognition of speech in a specified language, which can be changed at any time[1].

When it starts or the language is changed, the modality

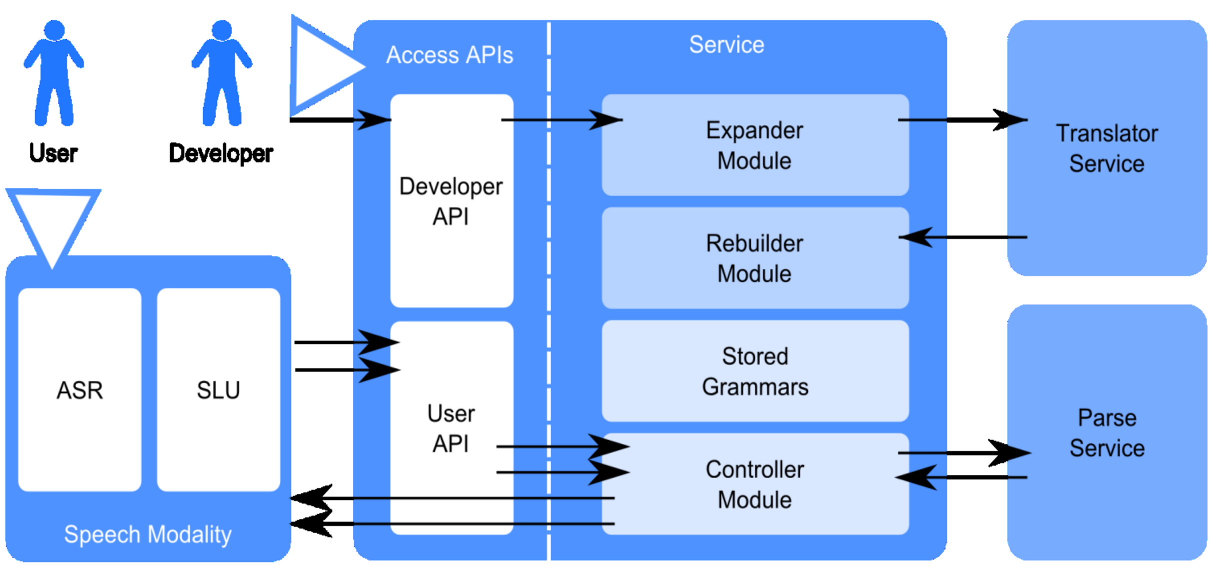


Figure 1. Overall architecture of the presented grammar generation service. The speech modality communicates with the core service, through specific APIs and external services are used for translation and parsing.

[1] It is mandatory that the speech engine and the support for the selected language are installed.

requests the service for the corresponding grammar, identified by an id and the desired language. The service answers with a GRXML grammar, which is directly loaded to a speech recognizer (ASR) engine and then waits for the user speech input. Grammars can be updated by a dynamic rule, which is the task addressed by the service described in the paper.

Each time the user speaks and a sentence is recognized by the speech engine, if the parameter of confidence of the recognition is bigger than a configured threshold, the modality requests the service to process that sentence in the SLU interpreter. The SLU extracts the semantic information of the sentence and sends it back to the modality, which relays its contents to the application through the interaction manager. Although grammars in the modality are in GRXML format, the translation service requires Phoenix grammars [24][25].

The Phoenix grammar is divided into frames containing slots and all slots start with a name and end with a semi-colon. The name of the slots is between square brackets. Inside slots any item string is between parentheses. Items can be of three types, the actual words to be recognized, other slots names, and variables. The information between parenthesis define the patterns for filler strings that can be accepted and are converted by Phoenix into recursive transition networks (RTNs) and are equivalent to context free grammars [19]. These patterns can also be seen as rules.

The difference between slots and variables are that variables are defined inside a slot and can only be used inside that slot. Finally, asterisks define an optional word. Fig. 2 is an example of the construction of a phoenix grammar.

```
[SlotName]
   (word1 word2 [other_slot] VARIABLE1)
   (word3 VARIABLE2 *word4 VARIABLE1)

 VARIABLE1
   (word5)
   (word6 word7)

 VARIABLE2
   (word8)
   (word9)
 ;
```

Figure 2. Example of a Phoenix grammar, From: From:http://wiki.speech.cs.cmu.edu/olympus/index.php/Phoenix_Grammar Reference

The construction of the grammar will be presented in Section IV, Subsection B – 2).

2) Main Service

The main service is responsible for the manipulation of the grammar. It allows: a) uploading files and input to be analyzed, and retrieving parsing results; b) getting all sentences generated by the specified grammars and on-demand translation of grammars; c) submitting corrections to derived grammars; and d) retrieving a list of all available grammars.

Overall, the service supports three usage scenarios. The simplest, illustrated in Fig. 3, consists in the submission of a grammar and the selection of an intended language, which results in the subsequent generation of a different GRXML grammar, supported by speech modality.

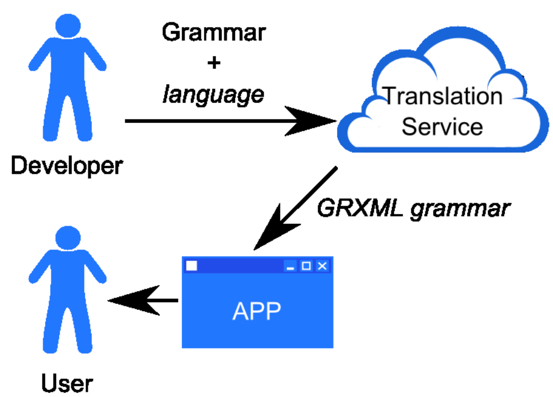


Figure 3. Simple use of the service to get list of sentences for ASR in a target language.

Fig. 4 shows a case where, assuming previous configurations and a working ASR, the service is used to extract semantic tags of a given text and return them to the caller. This way of using the service implements the multilingual SLU processing.

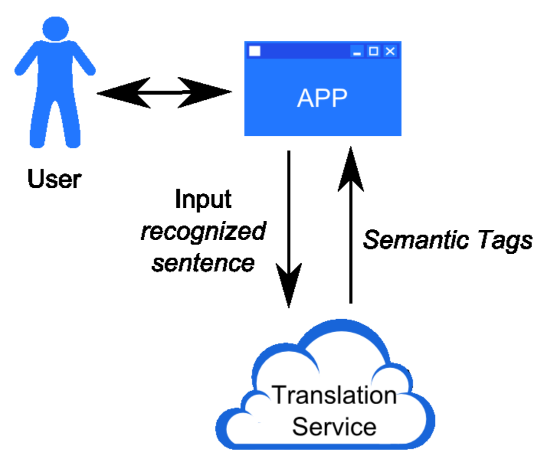


Figure 4. Service used as multilingual SLU.

Given the limitations of automatic translations, the service also supports manual revision and subsequent update of grammars (Fig. 5). This use is particularly suited when developing an application – such as AALFred – allowing the creation of an initial semantic grammar in English and using the service to provide translated grammars in other languages, enabling each involved partner in the project to revise and correct the automatically generated grammars. Each revised version becomes part of the service, after upload, and is used as described in the previous use cases.

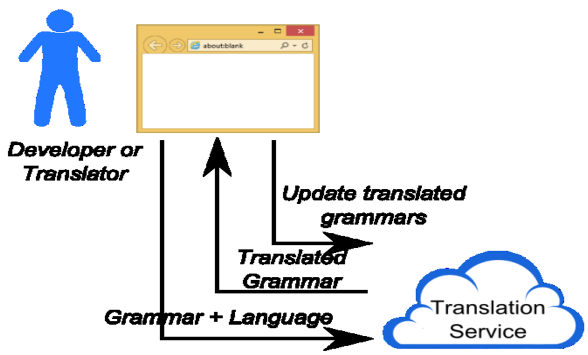


Figure 5. Service used to manual revision and update of grammars.

After the translation is accomplished, a Rebuilder Module recreates new grammars according to the translated languages. Afterwards, these new grammars are stored within the Stored Grammars module for further usage.

3) Access APIs

All operations are made through the access APIs, ensuring a consistent and complete operation control.

To enable the insertion of new grammars, a specific interface is required for the developer. This interface can be seen as a frontend, which allows the developer to submit a grammar and check the results of grammar translation, both in terms of generated grammar and of generated sentences. In our current implementation, it supports editing a grammar and its resubmission. This method enables faster feedback cycles of grammar enhancement.

For the speech modality, a user API is provided, allowing sequences of words from the speech recognizer to be processed in order to obtain semantic tags (i.e., to perform SLU in speech recognizer output).

4) Parser and translator services

The service is connected to two external services. The first one provides parsing of the word sequences resulting from the speech recognition process; the second provides translation of sentences.

IV. SERVICE IMPLEMENTATION

To test our architecture and associated ideas, a service has been created and used. Phoenix [24] was chosen as both the parser and grammar specification format. The advantages of this choice are explained by Phoenix's robustness to errors in recognition and parsing abilities. For translation, the choice fell on Bing due to its ability of providing reordering information. Later on, a more detailed explanation will be given on this.

The following sections provide information on the implementation and features of key components within the prototype.

A. Parser service

The objective of the parser is to extract the semantic tags, as defined in the semantic grammar, from the list of words received from the ASR, and return the text plus the semantic tags to be processed by the Interaction Manager and ultimately used by the application.

Internally, the analysis is done by Phoenix. Phoenix uses an automatic translated semantic grammar that allows tags existing on the original grammar to be preserved on the target language grammar.

In order to have an integrated support for the multiple languages of the project – or even other languages – the SLU parser is coupled with the management and process of automatic derivation of grammars by automatic translation.

B. Translation of Semantic Grammars

The goal is to translate to a target language all the terminal words while preserving the semantic tags. Translation must also produce a complete list of sentences defined by the grammar.

The process adopted and implemented is composed of three steps: 1) full expansion of the grammar; 2) translation; and 3) grammar rebuild.

1) Grammar Expansion

In order to be able to manipulate the Phoenix Grammar, one of two approaches had to be followed: either change the Phoenix Parser or have a separate parser to parse the Phoenix grammar structures onto a separate data structure, on which we would then apply our modifications. We decided to implement a separate parser so as not to change the Phoenix code, allowing us to use C# for our work and rely on the Phoenix Parser only for its already defined and well-tested function: parsing the input text based on a defined grammar.

In order to properly translate the grammar to take in consideration word reordering, we need to submit the full sentence to the translator. While a word-by-word translation would yield a non-natural result, submitting the whole sentence allows us to retrieve a translated sentence that sounds natural and takes in consideration language specific connectors and variances which may not exist on the original language.

To enable this sentence based translation we need to obtain all the sentences represented by the Phoenix grammar. This is done by a complete expansion of the grammar, replacing all the non-terminals by all the possibilities.

Without entering into code details, not the aim of this paper, the expansion is made by using a recursive method that makes use of three data structures: a list with all the rules that constitute the grammar, named **remainingList**; an

inProgress stack; and a **doneSoFar** queue. At the start, the grammar is parsed, the list of all rules created and the recursive expansion method invoked with this list as **remainingList** and an empty **doneSoFar** queue. The recursion starts by identifying the MAIN rule of the grammar and processing it.

Processing of a rule consists essential of: (1) adding, to the **remainingList,** the lines including non-terminals; (2) calling again the method to process the next terminal or non-terminal in the rule in processing; (3) if a terminal symbol is detected it is added to the **doneSoFar** queue; (4) if the symbol to be processed is a non-terminal, possible replacement values are added to the rule in processing and a recursive call is made.

During the expansion of all the rules, the history of the rules visited along the expansion is kept, and used in the grammar rebuilding process (step 3, explained ahead).

2) *Translation*

The translation process consists in submitting the result of the expansion and receiving the resulting translated sentences and the information regarding the pairing of words in the translation with the correspondent words in the source.

In our prototype, we selected Bing Translator as the translator service. The usage of the Bing Translator is an advantage to us since it provides the realignment info [26] necessary to get word reordering support during the grammar rebuild process. That realignment info both eases the matching of translation with source words and is what allows us to support word reordering when reconstructing grammar rules. In addition, Bing Translator also allows us to obtain multiple translations per request, which enables the expansion of an existing grammar to support several similar sentences, with no need of additional input by the developer. We can thus increase the coverage of our grammar in an automatic and effortless way.

The translation of a sentence obtained in the grammar expansion process is illustrated in Fig. 6.

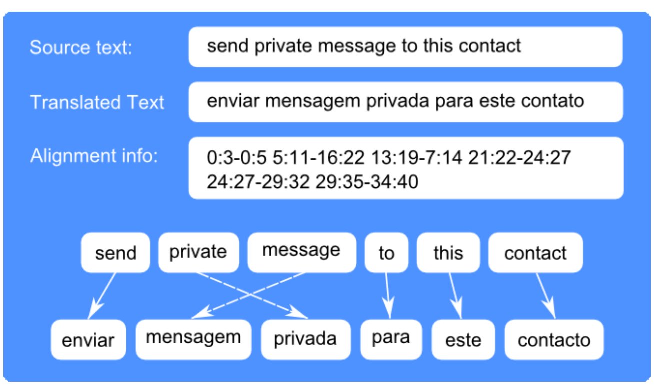


Figure 6. Example of the translation process of a sentence obtained by grammar expansion. In this case, the original sentence (Source Text) is in English and was translated using Bing to Portuguese (Translated Text). Additionally to the translated sentence is provided information on the alignment, which, for easier interpretation, is represented graphically at the bottom of the figure.

3) *Grammar Rebuild*

When the grammar is parsed (in order to expand it afterwards), a different object is created for each instance of any rule. As such, for each Terminal word present in the statement resulting from the expansion of the grammar, we can determine exactly which rule gave origin to the path that lead to it after the sentence is submitted for translation. Since we have reordering info available, we know which rules generated the text resulting from the translator.

The developed algorithm uses the saved Grammar Expansion history and the translated sentences of the Translation Process. It consists in analyzing the ancestors' history information to remake the grammar. This is done by merging Non-Terminals of the same level throughout the grammar in a top-bottom approach. Figs. 7 and 8 show an example. Fig. 7 presents, in 2 columns, the input grammar in English and the resulting Portuguese grammar. At the bottom of this figure are presented examples of the sentences resulting from grammar expansion (at left) and the sentences obtained by translation. How the translation results are used to create the translated grammar is illustrated in Fig. 8. After obtaining the representation shown in the figure, duplicates are eliminated automatically, thus obtaining the grammar according to the translation given.

[Main] ([AGENDA]) ; [AGENDA] (*view [WEEKDAYS] *schedule) ; [WEEKDAYS] ([MONDAY]) ([TUESDAY]) ; [MONDAY] (monday) ; [TUESDAY] (tuesday) ;	[Main] ([AGENDA]) ; [AGENDA] (*ver *calendário [WEEKDAYS]) ; [WEEKDAYS] ([TUESDAY]) ([MONDAY]) ; [MONDAY] (segunda-feira) ; [TUESDAY] (terça-feira) ;
monday tuesday monday schedule tuesday schedule view monday view tuesday view monday schedule view tuesday Schedule	segunda-feira terça-feira calendário segunda- feira calendário terça- feira ver segunda-feira ver terça-feira ver calendário segunda-feira ver calendário terça-feira

Figure 7. Example illustrating part of the grammar translation process, showing the input grammar in English, at left, and the resulting Portuguese grammar. At the bottom of the figure are presented examples of the sentences resulting from grammar expansion (at left) and the sentences obtained by translation.

[Main] [AGENDA] Ver	[Main] [AGENDA] Calendário	[Main] [AGENDA] [WEEKDAYS] [MONDAY] segunda-feira

Figure 8. Illustration of the process of grammar reconstruction for a sentence of the example presented in previous figure.

C. Dynamic Rules

In some cases, we might also need to have dynamic content in the grammars. For example, if we want to be able to select contacts by their names, we must allow a way to inform the system of the name of a new contact.

These dynamic rules must be created as all the other rules, but the developer does not provide the complete information for these rules. Also, they are not translated. These rules remain empty until the application needs to recognize dynamic content. When this happens, the application requests the service to update the current grammar. To do it, the application sends the identification of the grammar, the rule and the list of sentences to be inserted in that rule. After that, the service processes the new grammar and provides an updated version of the GRXML grammar for the Speech Recognition.

Fig. 9 illustrates the messages between application and service.

D. Manual Revision of the Translated Grammars

As automatic translation is not always accurate, the possibility of manually editing the automatic translation results was made possible by creating a website for revision to the grammars (Fig 10 shows a screenshot of the website). The process of revision is very simple. The translator can access a grammar and then choose the language to review. In the review page, there is the possibility to generate all the possible sentences and analyze the correctness of the sentences. Having identified errors in sentences, it is easy to find and correct the grammar. To simplify the process, translated grammar and original appear side by side. In addition to correct sentences, the human editor can also delete or add new possible sentences since they do not delete or change the rule names. At the end, before storing the changes, the grammar is validated to confirm that there are no syntax errors.

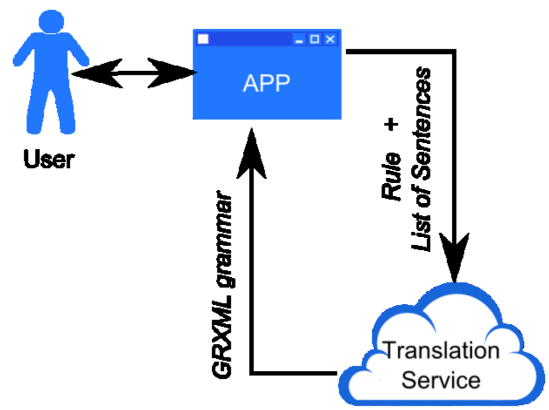


Figure 9. The messages between application and service when using the process of dynamic update of a grammar rule. First, the application sends a list of sentences or words to be added to a specified rule; after this, the service recreates the Grammar for the speech recognizer and sends it to the application.

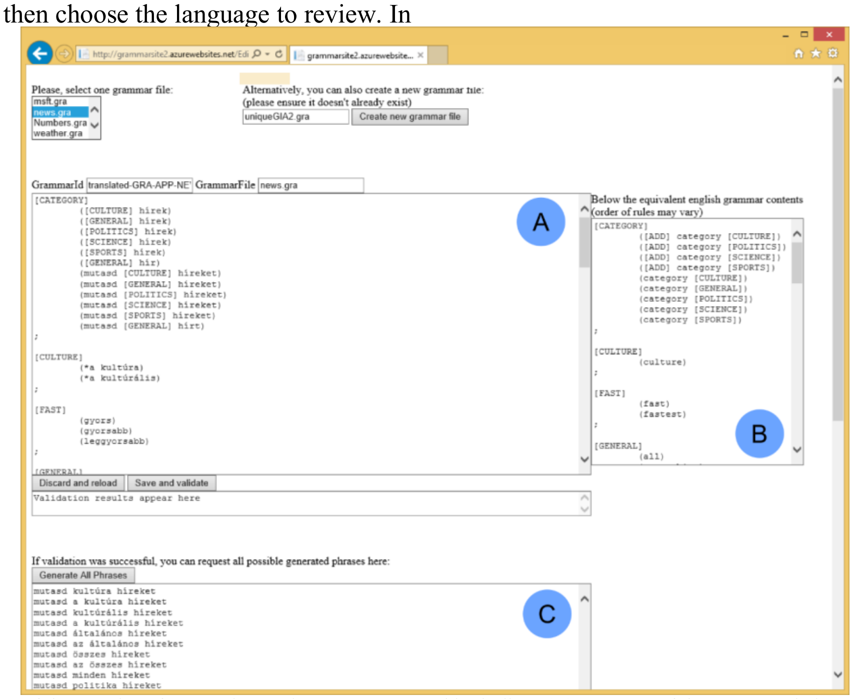


Figure 10. Screenshot of the user interface used to review grammars and including: (A) contents of the grammar being revised; (B) contents of the reference (English) grammar; and (C) a thorough list of all possible phrases generated based on the grammar in (A) for verification purposes.

V. Results

Currently, the developed service module supports the translation of text from English to French, Hungarian, Polish, and Portuguese. Furthermore, it supports translations from French, Hungarian, Polish, and Portuguese to English.

We start this section by presenting some illustrative examples of the service usage. Further on, we present information regarding real results of the adoption of this service for the multimodal multilingual Personal Assistant AALFred.

A. Representative Examples of Use

Illustrating service usage, three examples are presented for: (1) grammar translation; (2) grammar manual revision; and (3) dynamic definition of part of a grammar.

1) Example of grammar translation

After the submission of a new grammar to the service, it will be parsed and stored in memory after which all phrases will be generated. As an example, the grammar in Fig. 11 is converted to the Hungarian translation presented in Fig. 14.

As can be seen following the steps, the grammar in English is used to generate all sentences (Fig. 12), which are then translated. The translation (Fig. 13) is then used, in conjunction with word generation history, to rebuild the grammar in Hungarian, with flexibility to deal with word reordering (in bold) and to synonyms/alternatives (underlined).

```
[Main]
    ([ACTION])
    ;
[ACTION]
    ([GENERICENTITY])
    ;
[GENERICENTITY]
    ([NAVIGATION])
    ;
[NAVIGATION]
    (go [DIRECTION])
    (scroll [DIRECTION])
;
[DIRECTION]
    ([DOWN])
    ([LEFT])
    ([RIGHT])
    ([UP])
;
[DOWN]
    (down)
;
[LEFT]
    (left)
;
[RIGHT]
    (right)
;
[UP]
    (up)
;
```

Figure 11. Example of original grammar (in English) sent to the service by the developers of the News module of AALFred.

```
go down
go left
go right
go up
scroll down
scroll left
scroll right
scroll up
```

Figure 12. Result from the expansion of the original grammar (in English).

```
menjen balra
menjen felfelé
menjen le
menjen lefelé
menjen jobbra
balra görgetéshez
felfelé görgetéshez
le görgetéshez
lefelé görgetéshez
jobbra görgetéshez
görgetés balra
görgetés felfelé
görgetés le
görgetés lefelé
görgetés jobbra
menj balra
menj felfelé
menj le
menj lefelé
menj jobbra
lapozzunk balra
lapozzunk felfelé
lapozzunk le
….
```

Figure 13. Results from translation of the sentences in Fig.12 to Hungarian.

```
[Main]
    ([ACTION])
    ;
[ACTION]
    ([GENERICENTITY])
    ;
[GENERICENTITY]
    ([NAVIGATION])
    ;
[NAVIGATION]
    (menjen [DIRECTION])
    ([DIRECTION] görgetéshez)
    (görgetés [DIRECTION])
    (menj [DIRECTION])
    (lapozzunk [DIRECTION])
    (felmegy)
    ;
[DIRECTION]
    ([LEFT])
    ([UP])
    ([DOWN])
    ([RIGHT])
    ;
[LEFT]
    (balra)
    ;
[UP]
    (felfelé)
    ;
[DOWN]
    (le)
    (lefelé)
    ;
[RIGHT]
    (jobbra)
    ;
```

Figure 14. The resulting Hungarian grammar.

2) An example of grammar manual fine tuning

The system autonomously generates a grammar ready to be used on any language. However, it is possible to fine-tune the grammar to achieve a higher degree of correctness. This can be done by the developer or by a third party. The web based grammar editor allows previewing the sentences that the edited grammar describes and resubmission of the grammar. Each partner revised the translation of is language. In Fig. 15, the automatic translation and the human revised version of the previous example are shown for French.

```
[NAVIGATION]
        (défilement [DIRECTION])
        (aller [DIRECTION])
        (faites défiler [DIRECTION])
        (défilez [DIRECTION])
        (allez [DIRECTION])
        ;
```

```
[NAVIGATION]
        (défilement [DIRECTION])
        (aller à [DIRECTION])
        (faites défiler à [DIRECTION])
        (défilez à [DIRECTION])
        (allez à [DIRECTION])
        ([DIRECTION])
        (défilement à [DIRECTION])
        (défiler à [DIRECTION])
        (va à [DIRECTION])
        (vers la [DIRECTION])
        ;
```

Figure 15. Example of an automatic translated rule (at top) and its revised version (at bottom)

3) Example of dynamic definition of part of a grammar

In AALFred each user has different contacts and there is the need to recognize the name of the contacts. For that, the "NAMES" rule is updated in runtime. Fig. 16 presents the initial grammar (with only a placeholder) and the updated grammar with the names of contacts sent by AALFred (in this case the authors of this article).

`[NAMES]`
`(zxzxzxzx)`
`;`
	`[NAMES]`
	`(António Teixeira)`
	`(Pedro Francisco)`
	`(Nuno Almeida)`
	`(Carlos Pereira)`
	`(Samuel Silva)`
	`;`

Figure 16. Dynamic rule content; (left) initial grammar fragment; (right) updated grammar with names

B. Real Application Example - Supporting AALFred

The service presented was evaluated and continuously evolved in the development and field trials of a real application, the already mentioned Personal Assistant of Paelife project. With the proposed methods we aimed to provide developers with the tools needed to deploy applications supporting speech interaction in multiple languages. In this context, the achievements of project Paelife, involving a multinational consortium and supporting speech interaction in five different languages is, in our opinion, the most illustrative and significant result that can be reported.

1) AALFred modules

The development of AALFred has involved several partners, from different countries and languages, and each partner developed one or more modules. Each module provides users with a number of features enabling access to different content, from social messages, agenda, contacts, places of interest, news, and weather. Each partner created a fragment of a grammar to support the module. Table I shows examples of commands for each module.

TABLE I. MODULES INTEGRATING AALFRED AND EXAMPLES OF COMMANDS ACCEPTED FOR EACH ONE.

AALFred Module	Examples of commands integrating the grammar for the module [English version]
Messages	Open first message Delete selected message Replay to this message Send new email
Agenda	Open Monday Add new appointment Delete selected appointment
Contacts	Change photo Edit this contact Call him
Places of Interest	Find bus station Zoom in Find services near me
News	Category sports Open second item Read content
Weather	Five days forecast Show map Choose location

2) Grammar

In order to organize the grammar and have a structured analysis of the semantic output, we defined a format and structure which any developer needed to follow. By doing this, setting rule names and how the semantic output will be gets clearer. This structure also provides a unification of the semantic output from the various modules, making it easier for the application developer. The convention adopted is based on the Speech Acts [27][28].

For unification purposes, Main must be the first rule. It will consist of several rules that derive directly from Speech Acts, namely:

- **ACTION** – to specify all the actions in the grammar. Inside this rule must be specified all the other rules that

represent entities that can be affected by an action, like Contacts or Messages for example.
An action obligates the listener (in our case AALFred) to either perform the requested action or communicate a refusal or inability to perform the action.

- **INFORM** – used to communicate information to the listener. For example to specify the value of a setting in the app.
- **HELP** – exclusively reserved for the helping system that is being developed for AALFred. It will provide tips to the user based on his expertise on the app.

In most of the cases, an action will be referring to a specific entity. For example, delete can refer to a contact or a message. But, there are also some actions, like "OK" or "CANCEL", which do not refer to any entity in specific. Yet we have to associate them to an entity and that is where GENERICENTITY comes in. Despite its usefulness, we tend to avoid the use of GENERICENTITY as the implementation in the application is harder since it obligates to be aware of the context.

As mentioned, ACTION needs the specification of an entity. Examples of entities in AALFred context are: Contacts, News, and Pharmacies. Each entity has a number of attributes.

To provide the application developers with a simple and unified semantic output we adopted the following organization – with direct implication on the output obtained from the semantic parser - was adopted:

Main
→ Acts (Action | Inform | Help)
→ Entity
→ Parameters

Example of usage of the proposed convention: Considering the excerpt of grammar presented in Fig. 17, related to a news module. It allows the selection of the categories of the news. In this case, if the user says "category sports", the output of the semantic analysis is:

Main → ACTION → NEWS → CATEGORY → SPORTS

Meaning that an Action must be performed and that the entity is the News and the parameter is "Sports". Also, every action related to the news modules (an Entity) starts with Main-ACTION-NEWS.

```
[Main]
        ([ACTION])
        ([INFORM])
        ([HELP])
;
[ACTION]
        ([GENERICENTITY])
        ([NEWS])
        …
;
[NEWS]
        ([CATEGORY])
        …
;

[CATEGORY]
        (category [CULTURE])
        (category [SPORTS])
        …
;

[CULTURE ]
        (culture)
;

[SPORTS]
        (sports)
;
```

Figure 17. Excerpt grammar of the news module.

This convention is not mandatory for the use of the translation service as it works with any correct grammar, but revealed essential for the development of AALFred.

3) *AALFred semantic grammars*

Following the conventions presented in the previous section, the different teams of developers created the grammars for the spoken interaction of their modules. After, all the fragments were added together, resulting in a large grammar with a total of more than 250 rules, which were automatically translated for each language.

4) *On the translated grammars*

As native speakers of each of the languages to which grammars were translated made a manual revision, some insight on the quality and usefulness of the translation process can be obtained by comparing the automatically obtained grammars with the revised versions.

Fig. 18 shows the number of lines for the grammar in each language before and after the manual revision. The graphic shows that automatic translation inserts more sentences, with the same meaning, to the original (all languages have more lines that English). Portuguese and French revisers added more sentences in addition to those already in automatic translation and Polish and Hungarian have deleted some.

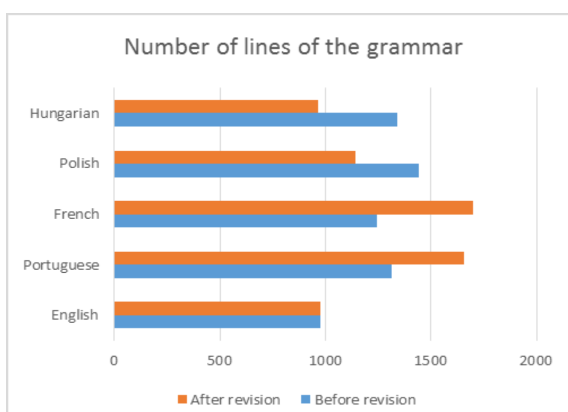


Figure 18. Number of lines for the grammar in each language before and after the manual revision.

In order to investigate in more detail, as human reviewers had the freedom to add and delete in any position, an automatic alignment process was applied to the pair of automatic translated and revised grammars, based on the number of changed characters and using the Levenshtein distance. A threshold of 20% of changes was adopted to decide between 3 classes: edited, deleted, and added. Also a 2 columns HTML output of the process results was created to allow manual inspection of the results. This HTML was used to define the mentioned threshold.

The result of this analysis is presented in Fig. 19. The graphic shows the number of new lines, number of lines that remained unaltered, number of lines that had some minor edition and number of lines that were removed.

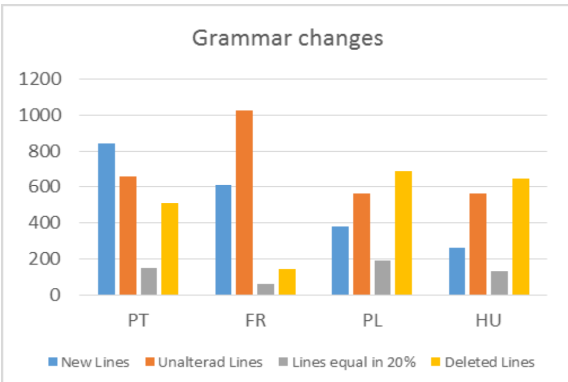


Figure 19. Information on the relation between the grammars obtained by automatic translation and their revised versions.

French was the language where more lines were accepted by the human reviewer. Also, while for French and Portuguese more than 500 lines were added, for Hungarian and Polish there were more deleted lines than new lines. This results combined suggest two different groups of system use. For one group the base grammar obtained by translation was used as a basis for extension; for the other to create a more correct one.

As the reviewers also reported that the revision process took a reasonable amount of time, all added, the results from the creation and translation process for AALFred were positive. Also, the PaeLife consortium decided for the usage of the grammars obtained and of the service to support SLU in the field trials in France, Hungary, and Poland. In the scope of another project, the Portuguese national project AAL4ALL [29], the Portuguese version of AALFred was also subject to field trials using the service described in this paper. The results of these evaluations will be the subject of forthcoming publications, some in preparation.

VI. Discussion

The results of the translation and manual revision of AALFred's grammars show differences regarding the target language used for translation. In a group of languages, human reviewers mostly added new patterns and made some corrections, enriching the grammar. For other languages, revision consisted mainly in removing incorrect patterns.

While these results could at first point to a very weak usefulness of the service, the feedback from developers and project partners point in a very different direction. In fact, for the application envisioned for the service, the many additions made were potentiated by the fact that an initial grammar already exists and reviewers could look for missing cases. Also some of the additions are enrichments to the original English grammar, adding patterns not initially included by the developers.

The fact that deletion of incorrect patterns was higher for languages such as Hungarian or Polish can eventually be related to a higher degree of complexity of the translation from English to those languages and/or lower performance of the translator.

For both groups of languages, the initial grammar produced by translation revealed itself as very useful.

VII. Conclusions and Future Work

Supporting multiple languages in spoken language interaction with machines is a complex task. In this paper is proposed the use of automatic translation applied to the generation of grammars for a set of target languages having as a basis a semantic grammar in English created by the application developers. Also, in the context of a generic service-based speech modality, proposed by the authors, a service is presented which aims to provide support for easy deployment of applications supporting several languages. The main highlight of the proposed service is the possibility to generate grammars for the speech recognition and SLU modules in different languages by automatic translation of an existing grammar (in English). A first prototype has been implemented, tested, and adopted in a Personal Assistant, AALFred.

Several examples of the service capabilities are presented. These examples are complemented with information regarding how the service is being used to support both development and real usage of AALFred.

Future developments should explore the use of multiple translation services, increasing the probability of having, in the set of translated sentences, the correct ones. Also, the service has potential to be used in new applications using this or other sets of languages.

Acknowledgments

Authors acknowledge the funding from AAL JP, FEDER, COMPETE and FCT, in the context project of AAL/0015/2009, project AAL4ALL (www.aal4all.org), IEETA Research Unit funding FCOMP-01-0124-FEDER-022682 (FCT-PEstC/EEI/UI0127/2011) and project Cloud Thinking (funded by the QREN Mais Centro program, ref. CENTRO-07-ST24-FEDER-002031).

References

[1] A. Teixeira, P. Francisco, N. Almeida, C. Pereira, and S. Silva, “Services to Support Use and Development of Speech Input for Multilingual Multimodal Applications for Mobile Scenarios,” in *The Ninth International Conference on Internet and Web Applications and Services (ICIW 2014),Track WSSA*

- *Web Services-based Systems and Applications*, 2014, pp. 41–46.

[2] T. H. Bui, "Multimodal Dialogue Management - State of the art," 2006, no. TR-CTIT-06–01.

[3] N. O. Bernsen, "Towards a tool for predicting speech functionality," in *Speech Communication*, 1997, vol. 23, no. 3, pp. 181–210.

[4] C. Munteanu, M. Jones, S. Oviatt, S. Brewster, G. Penn, S. Whittaker, N. Rajput, and A. Nanavati, "We need to talk: HCI and the delicate topic of spoken language interaction," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, 2013, pp. 2459–2464.

[5] A. Teixeira, F. Ferreira, N. Almeida, A. Rosa, J. Casimiro, S. Silva, A. Queirós, and A. Oliveira, "Multimodality and Adaptation for an Enhanced Mobile Medication Assistant for the Elderly," in *Proc. Third Mobile Accessibility Workshop (MOBACC), CHI 2013*, 2013.

[6] N. Almeida, S. Silva, and A. Teixeira, "Design and Development of Speech Interaction: A Methodology.," in *Proc. HCI International*, 2014.

[7] A. Teixeira, A. Hämäläinen, J. Avelar, N. Almeida, G. Németh, T. Fegyó, C. Zainkó, T. Csapó, B. Tóth, A. Oliveira, and M. S. Dias, "Speech-Centric Multimodal Interaction for Easy-To-Access Online Services : A Personal Life Assistant for the Elderly," in *Proc. DSAI 2013, Procedia Computer Science*, 2013, pp. 389–397.

[8] J. H. Christensen, "Using RESTful web-services and cloud computing to create next generation mobile applications," in *Proceedings of the 24th ACM SIGPLAN conference companion on Object oriented programming systems languages and applications*, 2009, pp. 627–634.

[9] M. Johnston, G. Di Fabbrizio, and S. Urbanek, "mTalk - A Multimodal Browser for Mobile Services.," in *INTERSPEECH*, 2011, pp. 3261–3264.

[10] "iOS - Siri." [Online]. Available: http://www.apple.com/ios/siri/. [Accessed: 14-Mar-2015].

[11] "Meet Cortana to Windows Phone." [Online]. Available: http://www.windowsphone.com/en-us/how-to/wp8/cortana/meet-cortana. [Accessed: 14-Mar-2015].

[12] "PaeLife: Personal Assistant to Enhance the Social Life of Seniors." [Online]. Available: http://www.microsoft.com/portugal/mldc/paelife/. [Accessed: 14-Mar-2015].

[13] A. Hämäläinen, A. J. S. Teixeira, N. Almeida, H. Meinedo, T. Fegyó, and M. S. Dias, "Multilingual speech recognition for the elderly: The AALFred personal life assistant," in *Proc. DSAI 2015, Procedia Computer Science*, 2015.

[14] "Voice Search." [Online]. Available: http://www.google.com/insidesearch/features/voicesearch/index-chrome.html. [Accessed: 14-Mar-2015].

[15] B. Singh, N. Kapur, and P. Kaur, "Speech Recognition with Hidden Markov Model: A Review," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 3, pp. 400–403, 2012.

[16] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the CMU-cambridge toolkit.," in *Eurospeech*, 1997, vol. 97, pp. 2707–2710.

[17] T. Brøndsted, "Evaluation of recent speech grammar standardization efforts.," in *INTERSPEECH*, 2001, pp. 1089–1092.

[18] A. Hunt and S. McGlashan, "Speech Recognition Grammar Specification Version 1.0." [Online]. Available: http://www.w3.org/TR/speech-grammar/. [Accessed: 14-Mar-2015].

[19] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[20] R. De Mori, F. Bechet, D. Hakkani-Tur, M. McTear, G. Riccardi, and G. Tur, "Spoken language understanding," *Signal Process. Mag. IEEE*, vol. 25, no. 3, pp. 50–58, 2008.

[21] D. A. Dahl, "The W3C multimodal architecture and interfaces standard," *J. Multimodal User Interfaces*, vol. 7, no. 3, pp. 171–182, Apr. 2013.

[22] A. Teixeira, N. Almeida, C. Pereira, and M. O. e Silva, "W3C MMI Architecture as a Basis for Enhanced Interaction for Ambient Assisted Living," in *Get Smart: Smart Homes, Cars, Devices and the Web, W3C Workshop on Rich Multimodal Application Development*, 2013.

[23] N. Almeida and A. Teixeira, "Enhanced interaction for the Elderly supported by the W3C Multimodal Architecture," in *5ª Conferência Nacional sobre Interacção*, 2013.

[24] W. Ward, "Understanding spontaneous speech: the Phoenix system," in *International Conference on Acoustics, Speech, and Signal Processing, 1991. ICASSP-91.*, 1991, vol. 1, pp. 365–367.

[25] "Phoenix Grammar Reference - Olympus." [Online]. Available: http://wiki.speech.cs.cmu.edu/olympus/index.php/Phoenix_Grammar_Reference. [Accessed: 14-Mar-2015].

[26] "Word Alignment Information from the API." [Online]. Available: http://msdn.microsoft.com/en-us/library/dn198370.aspx. [Accessed: 14-Mar-2015].

[27] H. Bunt, J. Alexandersson, J. Carletta, J.-W. Choe, A. C. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum, "Towards an ISO Standard for Dialogue Act Annotation," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 2010.

[28] S. Young, "CUED standard dialogue acts," 2009.

[29] "AAL4ALL | Ambient Assisted Living For All." [Online]. Available: http://www.aal4all.org/?lang=en. [Accessed: 14-Mar-2015].

AEQUO: Enhancing the Energy Efficiency in Private Clouds Using Compute and Network Power Management Functions

Kai Spindler*, Sven Reißmann†, Ronny Trommer*, Christian Pape*, Sebastian Rieger* and Thomas Glotzbach‡
*Department of Applied Computer Science, Fulda University of Applied Sciences, Fulda, Germany
{kai.spindler, ronny.trommer, christian.pape, sebastian.rieger}@cs.hs-fulda.de
†Computing Centre, Fulda University of Applied Sciences, Fulda, Germany
sven.reissmann@rz.hs-fulda.de
‡Elektrotechnik und Informationstechnik, Hochschule Darmstadt University of Applied Sciences, Darmstadt, Germany
thomas.glotzbach@h-da.de

Abstract—**Today's data centers need a huge amount of energy for their operation. Private cloud infrastructures using virtualization technologies are the prevailing paradigm in modern data centers and their energy consumption and the corresponding ongoing operational costs are not negligible. Solutions that raise the energy efficiency allow reductions in these operational costs and optimizations of the utilization of the data center infrastructure. Also, renewable energy sources can help to provide the needed energy, but usually these sources are fluctuating. Therefore, the energy is not always available when needed and also not always produced near the point of use. Further, the storage of energy is not available in industrial scale. The following article examines the possibility to shift the energy consumption of virtual machines and presents a lightweight prototype that can be integrated in private cloud environments using standard OpenStack components and application programming interfaces. It optimizes the energy efficiency by observing the current utilization parameters of compute resources and by taking appropriate actions based on this data. Furthermore, we evaluated mechanisms to control the energy efficiency of network resources. This optimization will be carried out by an automated instance, possessing a comprehensive view on the data center assets, which relocates virtual machines and optimizes the network structure. The article is completed with an evaluation to measure power consumption of data center assets during virtual machine live-migration operations and also illustrates further areas of research.**

***Keywords*–Private Cloud; Energy Efficiency; Renewable Energy; Computer Networks; Power Management.**

I. Introduction

Private or enterprise cloud solutions are currently gaining more and more momentum, mainly driven by the success of cloud-based services [1] and virtualization, but also by the ongoing eavesdropping scandals that hinder the use of public cloud providers for sensitive information. One of the major benefits of cloud-based services is formed by their scalability. This scalability is supported by the "elasticity" [2] of the underlying infrastructure that allows providers to support large-scale applications and services [3] for a vast number of mobile devices (e.g., smart phones, tablets) and users from all over the world. However, the improvement in scalability is achieved at the cost of larger data centers and growing energy consumption. Energy is not only needed to supply the IT infrastructure itself with electricity, but also for appropriate cooling. Hence, energy costs are one of the major challenges for current data centers.

Since cloud services are based on distributed systems, besides compute and storage, another essential resource is the network, enabling fast and decentralized access to the services over the Internet and especially the Web. This is also described as "broad network access" in [2]. To provide cloud and web-based services, efficient IT virtualization techniques and computer networks are necessary. These technologies in turn have an impact on the energy consumption and cost. Hence, adaptive power management based on the current requirements, i.e., the load on the applications and services, helps to increase the energy efficiency by turning components on and off or reducing their performance (e.g., throttling, energy saving functions). Such adaptive power management functions can also balance or consolidate the power consumption in private cloud environments. As cloud services are provided on an "on-demand" basis according to [2], an adaptive management based on the current load of the resources is supported by this major cloud paradigm.

In [1], we presented a solution to enhance the energy efficiency in OpenStack-based private cloud environments. This article elaborates on the implementation and concepts outlined in [1] and introduces a combination with renewable energy. A special focus is put on the efficient placement of virtual machines (VM) and the reduction of power required by network connections and components. Adaptive placement of VMs also permits a reduction of compute and storage power consumption by consolidating them on specific hosts, addressing the "resource pooling" requirement for cloud computing environments given in [2]. However, migration costs need to be considered. Hence, this article includes an evaluation of the power consumption of compute, storage and network components during VM migrations. A prototype that was implemented to monitor the energy efficiency (e.g., compute, storage and network utilization as well as temperature and thermal efficiency of the cooling) in cloud environments was presented in [1]. It includes throttling, enabling or disabling resources based on the current demand and given constraints (e.g., required fault tolerance, redundancy, quality of service parameters and network connectivity). The prototype uses standard cloud APIs (application programming interfaces) (i.e., OpenStack, Open Cloud Computing Interface (OCCI)). Therefore, it can easily be integrated in existing cloud infrastructures using standard OpenStack components.

The paper is laid out as follows. Section II gives an overview on private clouds based on OpenStack and describes the requirements for energy efficiency in such private cloud environments. Also, examples for existing techniques to enhance the energy efficiency in computer networks and references to related research projects are given. A major aspect of the research project behind this paper focuses on the use of renewable energy and to enhance the energy efficiency of distributed data centers. Therefore, Section III evaluates renewable energy fluctuation in Germany and defines requirements to leverage renewable energy sources for the energy-efficient use of resources in distributed data centers. Requirements for the implementation of our prototype, to enhance the energy efficiency by combining the state of the art techniques and extending them, are defined in Section IV. The implementation of our prototype and mechanisms to optimize the energy efficiency in private clouds are presented in Section V. Section VI describes an experimental testbed that was used for the evaluation of our concept and the implemented prototype, being presented in Section VII. Finally, Section VIII draws a conclusion, evaluates our research findings and outlines future work that will be pursued in the research project.

II. State of the Art

The following sections give an overview on the deployment of private clouds using OpenStack and examine the requirements for the energy efficiency of such environments. A special focus is drawn on the potential of energy-efficient computer networks. Additionally, related research projects are discussed.

A. OpenStack-based Private Clouds

The term cloud is an ambiguous concept and has been interpreted in many ways by vendors and customers of cloud services. One of the most sophisticated definitions is documented in NIST SP 800-145, expressing cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (i.e., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [2]. NIST identifies five essential characteristics, three service models, and four deployment models. Our work focuses on private cloud deployments with OpenStack, which is a software project that provides an open source implementation of technologies for building and operating public and private cloud environments using the "Infrastructure as a Service" (IaaS) service model. In OpenStack, this infrastructure is built by offering networking resources (named Neutron), compute resources (Nova), and storage resources, i.e., object storage (Swift) and block storage (Cinder). Additionally, OpenStack offers many more services for management and orchestration, such as Horizon and Heat, its identity service Keystone, and a telemetry service called Ceilometer.

The IaaS service model in OpenStack is implemented by providing VMs, which can run as Nova instances on the compute nodes of an OpenStack environment. The placement of VMs, being one of the main objectives of our work, can be on a specific Nova node or may depend on various parameters of the environment. Also, the migration of a running VM from one compute node to another, as well as the starting or stopping of VMs depending on the current load is possible during the lifecycle of a service. This flexibility provides some interesting aspects in terms of resilience (i.e., by seamlessly moving VMs from one data center to another) but also in terms of energy efficiency as we will demonstrate in detail later in this paper.

B. Energy Efficiency in Private Clouds

In today's rapidly-growing IT infrastructures, energy efficiency is no longer a secondary requirement, but has rather become one of the main objectives when planning and operating new data centers. One reason for this development is the common sensitization for an ecologically sustainable use of global resources. Furthermore, large-scale data centers consume enormous amounts of electrical power not only for running the IT systems, but also for cooling them. A measure for the ratio between the energy used by the computing equipment and the overall energy consumption of a data center is the power usage effectiveness (PUE), which takes into account, i.e., the energy needed for cooling and losses by (uninterruptible) power supplies [4]. At the same time, PUE has an impact on the operational overhead cost of a data center, hence its minimization is of great interest for today's data center operators, which have to act economical while facing increasing energy costs [5].

It can be said that cloud computing by definition leads to energy efficiency through its operational concepts, which include a better utilization of physical resources, dynamic scaling based on the current load, and location-independent and efficient resource management. However, to take advantage of these concepts, the whole cloud infrastructure needs to be carefully adapted to the operators' individual needs. For instance, resource pooling allows a cloud operator to consolidate multiple VMs providing various services on only a few physical hosts, hence increasing the efficiency of these hosts. At the same time, rapid elasticity and on-demand self-service concepts require the immediate and automatic availability of compute power if needed, therefore instant availability of additional resources is required [2].

The energy consumption of a VM running in OpenStack depends mainly on the energy requirements of its physical IaaS components, including compute (i.e., CPU (central processing unit), RAM (random-access memory)), storage (i.e., SAN (storage area network), NAS (network-attached storage), HDD (hard disk drive)), and networking components (i.e., routers, switches), but also on the distance between the components involved (e.g., the distance of the storage from the compute node). Consequently, the real power consumption ratio of a cloud service depends on the number of active compute, storage, and networking components needed to provide it. As VMs can be migrated from one physical host to another, it is possible to take advantage of fluctuating electricity prices or to adapt the load factor of a data center to climatic changes. This could be done not only by consolidating VMs in one data center, but also by sending the VMs to another geographical location, where operational costs are lower.

In OpenStack, the placement of VMs on a specific cloud computing fabric controller (Nova) is determined mainly by nova-scheduler [6]. While several techniques are offered for optimal VM placement, by default the so called Filter Scheduler is used. It supports the placement of a VM based on a physical location, available compute resources (e.g., CPU, RAM), or by its requirements for secondary resources, such as the availability of specific storage or network capabilities. Moreover, the Filter Scheduler addresses the operational requirements for resilience or consolidation of VMs by explicitly allowing a placement on different hosts or by grouping them on a single host. However, it does not take into account any energy efficiency parameters, neither for initial placement nor for the live-migration of VMs. Also, automatic migration of a VM in favor of load balancing or energy efficiency enhancements is not supported by nova-scheduler. Nevertheless, with its components for service orchestration (Heat) and telemetry (Ceilometer), OpenStack provides interfaces to manage VM migration that can be extended to evaluate energy consumption or cooling requirements.

C. Energy-efficient Computer Networks

Another aspect to take into account when measuring the energy consumption of a VM running in OpenStack is the networking equipment. Since nearly every current service or application is used over a network, the relevance of this aspect is rather obvious. In private clouds, the relevance is increased even further as cloud services are typically formed by a combination of different interdependent services in a data center. This can especially be observed in large cloud providers, where intra data center network traffic is several orders of magnitude higher than the traffic going out to the Internet [7].

From a theoretical point of view a network consists of multiple nodes, which are interconnected using links. Hence, a network and its resulting topology can be defined as a number of nodes and links. Looking at the power consumption, most of the links, especially in local area networks are passive, meaning that they do not consume individual power, but rather serve as a medium that carries electromagnetic or optical signals being generated at the nodes as a sender. Depending on the link characteristics (e.g., attenuation), signals might need to be refreshed, for example on long-distance links to allow the receiver to interpret the signal correctly. Optical and electromagnetic amplifiers can be applied to refresh the signal. Therefore, especially long-distance links can include amplifiers or special transceivers, e.g., directly attached to the cable. An overview on the cumulative energy consumption for the required amplifiers in long-distance optical networks is presented in [8]. In the model for the power consumption of a network, these transceivers and amplifiers can also be treated as nodes. In terms of power consumption these nodes are active components of the networks, as they individually draw power to refresh, send, receive and interpret the signals and the contained data. Nodes can interpret and modify the transfered data on different layers of the OSI or TCP/IP reference model.

The complexity of the protocols being interpreted in a network node, and the decapsulation necessary to get the corresponding protocol headers, are a major factor for the power consumption at the network nodes besides the power that is needed to send and receive the signals. Link characteristics (e.g., bandwidth, attenuation, length) define the power that is needed to send and receive the signals. All active components and their corresponding power consumption can be considered to enhance the energy efficiency of computer networks. Links have only an indirect influence on the power consumption. However, if links can be reduced, shortened or exchanged against media that support a higher energy efficiency, the power consumption of the network nodes can be lowered even further, though this is not always possible since locations of nodes and quality of links sometimes is implied by the physical location or local circumstances.

According to [9], computer networks typically account for 15–25% of the total energy consumption in data centers. The increasing number of users and the complexity of cloud services require a high bandwidth, which leads to increasing link speeds and, therefore, raises the power consumption of each switch port. This is also observed in [8], where the slope of energy consumption per bit at network devices like routers over the last years is lower than the slope of the continually increasing peak access bit rate. Accordingly, the total power consumption of networks is still increasing due to the increasing access bit rate and number of users regardless of improved networking equipment that consumes less watt per transfered bit. An overview on energy-efficient data center networks is given in [10]. Redundant links are required to assure resilience of the network, again increasing the power consumption. Because of the increased power consumption for higher bit rates and the number of redundant links, some researchers [11][12] already claim that the fraction of the energy consumed by the network in a data center is likely to rise to up to 50% in the near future. Concepts like Equal Cost Multipathing (ECMP) or Multipath TCP are available to utilize the equipment and redundant links up to the maximum capacity of the networks.

As today's networks are mostly not energy proportional [11][13], higher utilization of the network and its equipment leads to an increased energy efficiency of the network. However, variable bandwidth requirements (e.g., decreased utilization during nighttime) makes it economically reasonable to scale down the network as well [14]. For wired local area networks (LAN), which we primarily focus on, there are already some power management techniques being offered by network equipment providers. First and foremost, the LAN standard 802.3 was extended in 2012 to include 802.3az, also called Energy Efficient Ethernet (EEE) [15]. Since this extension is part of the regular 802.3-2012 standard, it is likely that in the near future all Ethernet equipment will support EEE. However, EEE was specifically designed for copper-based network links. With increasing bandwidth requirements, most links especially in the aggregation or core layer use optical links. While it is currently not possible to lower the link speeds of fiber physical transceivers (PHY) to reduce their energy consumption [16], the transceivers and hence optical links can be powered off if currently not needed [17]. Compared to copper PHYs, which support an idle state leveraging EEE, fiber PHYs unfortunately do not support an automatic wakeup, being related to the missing capability to lower the link speed [18]. Therefore, the power management of optical links currently needs an external power controller or network management system.

While network equipment manufacturers who include EEE in their products claim that 802.3az allows a reduction of the energy consumed by a single copper port by up to 81% [19], this benefit comes with the price of increased latency during the low power idle (LPI) phase [20]. Regarding the fact that currently data center network infrastructures are moving to 10 Gbit/s Ethernet and beyond, where power consumption per port is usually over 5 Watts [19], the power savings for the entire data center infrastructure are even higher. Furthermore, there are other vendor-specific power management functions of networking components (e.g., Cisco EnergyWise [5]) that are not covered by EEE. Also, as mentioned above, network power management techniques could be improved by temporarily powering off unused optical links or network functions. Unfortunately, such techniques also have a negative impact on the latency due to the power management and necessary wakeup cycles. Compared to power management functions of compute and storage resources (e.g., Advanced Power Management (APM), Advanced Configuration and Power Interface (ACPI)) that have constantly evolved over the last decades, power management functions for network components are relatively new and supposedly need to be improved due to energy efficiency requirements in the near future [12].

Existing solutions for energy-efficient networks concentrate on the reduction of the local power consumption on individual network components and ports, but they are typically unaware of the current global requirements in the entire network, especially when multiple network equipment providers are used. Therefore, their scope is rather limited and the energy efficiency optimization is rather isolated. As networks are a fundamental building block of private clouds and since "broad network access" is also an essential characteristic of cloud services [2], a holistic approach to enhance energy efficiency in private clouds should take all three cloud infrastructure components (compute, storage and network resources) into account.

Some research projects, notably Stanford's ElasticTree [21] have identified this problem, but did not integrate it with an appropriate placement of VMs and especially did not discuss the requirements of private clouds. Also, these solutions only state that switching off network components and their functions might be an option, but do not leverage or present corresponding network power management functions.

By using a network controller in private or enterprise clouds that is aware of the entire network topology, such network power management functions could be implemented for example to disable currently unused links or to throttle link rates during off-peak times while still maintaining fault tolerance requirements, e.g., in multipath network environments. Additionally, such a controller could enable power saving modes, standby or disable networking functions to lower the power consumption of the nodes. Existing network power management, as described above, could be combined with this approach. Moreover, such a controller could also activate and deactivate entire networking components based on the current requirements to enhance the energy efficiency. Hence, energy-efficient computer networks could reduce the power consumption of the network nodes (and the number of active links) to the minimum, especially in off-peak times while still ensure fault-tolerance and high performance under load. Possible solutions are presented in the forthcoming sections of this paper.

D. Related Work

Energy-efficient placement of virtual machines in OpenStack private cloud environments is also discussed in [22][23][24]. However, these approaches do not consider an optimal placement of VMs with respect to temperature, cooling and network connectivity requirements. Furthermore, these publications focus more on the evaluation of different algorithmic approaches for an optimal placement of VMs while keeping the cost of migrations low, than on the integration, thereby this paper will not go into detail on the evaluation of different algorithms. Additionally, the extensions presented in these papers cannot be used with the current Juno (nor the previous Icehouse and Havana) Release of OpenStack.

A more generalized evaluation of an energy-efficient placement of VMs in cloud environments and relevant parameters is given in [25] and [26]. However, these contributions do not offer testbeds for OpenStack environments. A tool that allows for distributing virtual machines considering the migration cost is introduced in [27]. It includes a basic analysis of migration cost and the impact of live-migration for an application. The CÆSARA project [28] outlines an algorithm for the energy-efficient placement of virtual machines. Basic concept is the estimation of a server's energy consumption based on the running virtual machines' characteristics. Furthermore, a distributed algorithm used for virtual machine placement in large cloud environments is discussed in [29]. The idea here is that every server knows the CPU load of the other physical servers. Each server tries to comply with an upper and lower threshold for the CPU load and initiates the migration of virtual machines when these thresholds are violated. Also, bin packing algorithms that form the basic concept of virtual machine placement on physical servers are still subject of current scientific studies and research [30][31][32].

Our research also highlighted the lack of studies examining the relationship between energy consumption and communication distance. Instead, merely average estimations are determined in the form of energy consumption per download quantity. In [33], a holistic view on energy consumption of network transactions including also the embedded energy resources used to manufacture network devices is given. The focal point of this publication are transmissions to end customers (e.g., including Digital Subscriber Line Multiplexers (DSLAMs) and telephone lines). Excluding these costs, the energy demand stated in this study is 149 Wh/GB for embedded energy and 849 Wh/GB for the real transmission, so in sum 998 Wh/GB $\approx$ 0.1 kWh/GB. Similar values were determined in [34] by measuring transmissions at an international conference between Switzerland and Japan. In this publication, the authors relied upon pessimistic assumptions, so a realistic value of 0.2 kWh/GB was postulated, thus, a higher value than in other studies. Furthermore, a comprehensive study from the year 2009 by the German OFFIS institute [35] identified possible savings by load management across multiple data centers and forecasted an energy demand of 0.1 kWh/GB for the year 2014, which corresponds with recent studies.

Concerning energy-efficient computer networks, especially the ElasticTree project [21] presented interesting starting points and related work for power management and throttling of network components using OpenFlow. The ideas of ElasticTree were extended, e.g., in the ECODANE project [36] to include traffic engineering. Also, theoretical energy-aware optimizations of data center networks were presented in [37][9]. Requirements and constraints for energy-efficient placement of VMs regarding the network connectivity, were explored in [38][39][40]. However, these solutions do not include existing power management techniques like the ones we described in the previous sections for networking resources (e.g., [9][19][20]). Furthermore, these approaches do not include power management functions like ACPI and related solutions. In our work, we combine the existing power management mechanisms and the solutions that were discussed in the related work given in this section, and present a lightweight extension to leverage power management techniques in existing OpenStack enterprise clouds.

III. Using renewable energy to reduce the power consumption of distributed data centers

While data centers (DC) usually need a lot of energy for their operation, environmental compatibility plays an increasingly important role. A number of key figures, like the already mentioned power usage effectiveness (PUE), as well as the data center infrastructure efficiency (DCIE) and carbon usage effectiveness (CUE) [41] need to be considered, when planning new or optimizing the efficiency of existing data centers. A number of certification programs are available, to provide an incentive for organizations to build efficient IT infrastructure. As an example, "Der Blaue Engel" (The Blue Angel) [42], a well known quality seal for environmental compatibility assigned by the German federal environment agency has recently enlarged its certification program for data centers. Criteria for the award include appropriate PUE monitoring, application of efficient hardware components, as well as meeting most of the data centers electricity demand from renewable energies, such as hydroelectric power, photovoltaics (PV), wind power, biomass energy, or from combined heat and power generation plants. In the future, it is conceivably that legal obligation will force large data centers to meet some of the requirements of such certification programs. As a result, an energy supply with the help of the renewable energy sources would be eligible. However, from the perspective of the energy suppliers one problem occurs.

The energy output of renewable energy sources is fluctuating. That means the energy is not always available when needed or vice versa. In addition, the energy is not always produced near the point of use (e.g., offshore wind energy [43]). First of all, this leads to the necessity to store the energy in between [44] or shift the consumption in time [45]. Until now, the storage of energy is just conditionally feasible, as it is expensive and not available in industrial scales. In contrast, the possibility to shift the energy consumption of data centers with the help of an intelligent energy management is viable. Regarding wind energy it is obvious that in the case of heavy winds at the shore, the produced energy has to move via overhead lines, which means that the power grid has to be extended and rebuilt in future [46]. While in the classical approach, the energy has to be moved from the power plant to the location of the data center, the following solution is focusing on the other way around. The services of the data center as consumer will be moved to the place of energy production. This is supported by the increasing interest in server, storage and network virtualization like software-defined networking, supporting current IT and cloud infrastructures. Therefore, migrations should be as fast and cheap as possible to benefit from the advantages of fluctuations in renewable energy sources and related savings of operational costs of local distributed data centers.

Figure 1 shows an overview of an intelligent energy management in data centers with software-defined networks and renewable energies. Due to the current demand of the shown DC 2, not all available IT resources are actively used. Inactive components and connections are marked with dashed lines. If an energy surplus arises due to strong winds and available onsite wind turbines, inactive components can be immediately activated and virtual machines (virtual resources) or applications can be transferred from DC 1 to DC 2. Conversely, the virtual machine can be shifted from DC 2 to DC 1 if no wind energy is present but energy is supplied by photovoltaic power plants. Current server virtualization and infrastructure solutions enable this migration of IT resources in the background. As DC 2 uses a different Internet connection, the only noticeable difference to the users of the resettled services might be a higher or in ideal case a lower latency related to the new distance between user and service. The DC 2 uses a different Internet connection. Related to the new distance between user and service is just a higher or in ideal case a lower latency noticeable while using the service. Within Germany, these latency changes are generally below 50 milliseconds. Initially, a time series analysis of meteorological data has been carried out over three years' data to determine the potential of shifting services from one DC to another.

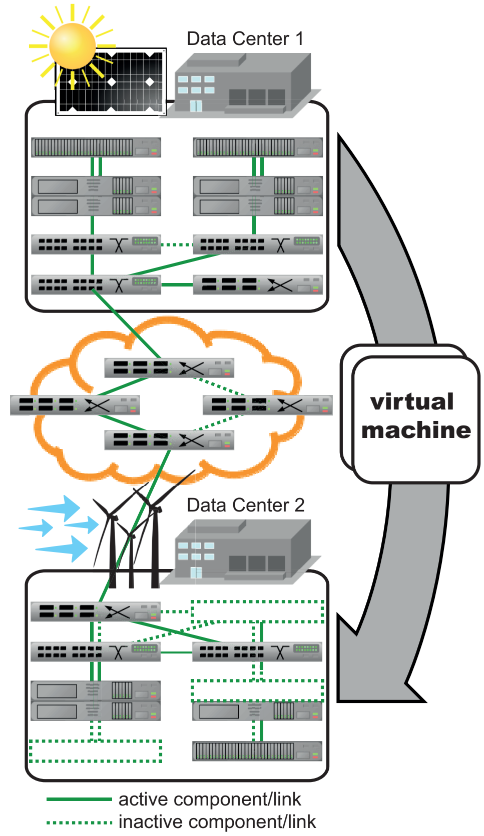


Figure 1. Benefits of renewable energy and a proper energy management for data centers.

The meteorological data used was derived from the years 2011 to 2013 at three different locations. The weather conditions in terms of renewable energy depend mainly on the latitude. Therefore, the locations Cuxhaven (northern Germany), Frankfurt am Main (in the middle of Germany) and Munich (southern Germany) were picked. From these locations, measured values were used with an interval of 15 minutes. Overall, 105.120 individually measured values per time series were evaluated. The most important short and medium term sources for electrical power supply from renewable energies are wind and solar energy (photovoltaic). For this reason, the focus is put on three indicators for wind speed in m/s, the irradiance in W/m^2 and the ambient temperature in °C. Based on these indicators, it is possible to calculate the power

TABLE I. Overview of the installed capacity of wind and photo-voltaic power divided into northern, central and southern Germany [47][48].

Separate				
Region	Wind [MW]	Wind [%]	PV [MV]	PV [%]
Northern Germany	17,305	44.2%	6,289	16.4%
Central Germany	16,854	43.0%	13,550	35.4%
Southern Germany	5,005	12.8%	18,402	48.1%

Total		
Region	Wind	PV
Northern Germany	73.3%	26.7%
Central Germany	55.4%	44.6%
Southern Germany	21.4%	78.6%

(power time series) for wind and photovoltaic energy, using mathematical methods. Due to the existing north-south divide of wind speed and irradiance, there is a certain distribution of power plants. The wind speed has a strong increase from south to north. The biggest values of wind speed are measured on the German coast in the far north. Therefore, the biggest percentages of wind turbines are located in the north and in the plain regions of central Germany. The situation in photovoltaic is exactly the opposite. In southern Germany, you will find a larger irradiance and for this reason a bigger energy yield. Thus, most photovoltaic power plants are based in southern Germany. Table I shows an overview of the installed capacity of wind power and photovoltaic power divided into northern, central and southern Germany [47][48]. The calculated power time series of the two energy sources were converted into percentage values to make a conclusion out of current power distribution of the three places. The determined distributions of Table I were included.

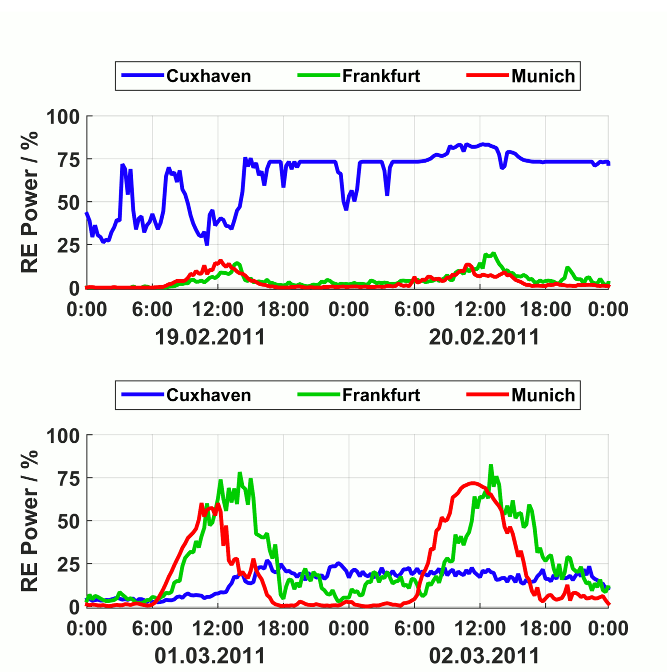


Figure 2. Percentage power outcome of renewable energies on four days.

Figure 2 shows an example of the daily distribution of power. In the upper chart it can be seen, that Cuxhaven has a lot bigger outcome of renewable energy sources than at the two other sides. At this time, the services of the DCs in southern and central Germany should be shifted to the DCs in northern Germany. In the lower part of the figure it is shown that, at least during the day, more power in Frankfurt and Munich is available. In that case a corresponding shift should occur.

TABLE II. Times when a displacement would be possible (data in hours and the percentage of total time (26,280h)).

From \ To	Cuxhaven	Frankfurt	Munich
Cuxhaven	-	581h (2.2%)	1,454h (5.5%)
Frankfurt	2,654h (10.1%)	-	1,093h (4.2%)
Munich	2,971h (11.3%)	771h (2.9%)	-

In Table II, the evaluation of the whole time series over three years is shown. In each case, two locations were compared to each other to calculate the sum of time in hours, to find a situation where a displacement would be possible. Table II shows the maximum possible hours and the percentage of total time (26,280h) when a displacement would be possible (best case scenario). This scenario always occurs if the difference between the current powers at the two sides is greater than or equal to 30%. A great potential becomes visible. The next step is now to show if and when a displacement is possible and reasonable. A Mathlab/Simulink simulation, which is under construction, should provide information about the potential of shifting DCs. These simulations will consider models of complete DCs and use official weather data to include renewable energy sources in the evaluation.

IV. Energy-efficient Placement and Network Connectivity of Virtual Machines

In the following sections, we describe various capabilities of OpenStack regarding the placement of VMs and identify requirements for adding energy efficiency criteria to this process. A special focus is laid on the energy efficiency of the network connection between VMs in distributed private clouds.

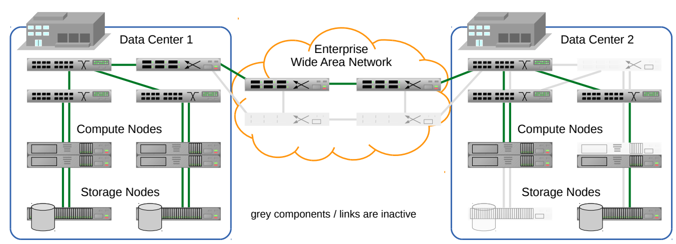


Figure 3. Power management for energy-efficient compute, storage and networking resources in private clouds.

Figure 3 shows an example of a private cloud IT infrastructure that is distributed over two data centers at different sites. Each data center provides compute, storage and network resources as described in Section II-A. Regarding the power management, each of these components consumes energy based on its utilization. Furthermore, as the components are connected to each other over the network, by deactivating or throttling individual components or links, the energy consumption of the private cloud can be reduced, e.g., during off-peak times. Also, redundant components or links can be deactivated completely in favor of increased energy efficiency when active fault tolerance is not needed, e.g., due to low utilization. The deactivation or throttling is symbolized by the grayed out links and components shown in Figure 3.

A. Energy-efficient Placement of Virtual Machines in OpenStack Environments

As described in Section II-A, OpenStack is not, by itself, able to manage resources with respect to energy efficiency. Therefore, we present concepts to support the decision-making process about when and how resources like VMs can be relocated to increase the energy efficiency with respect to the required dependencies (i.e., storage, network). To decide whether or not to move a VM from one host to another, it is necessary to know various metrics about the system that runs the hypervisor. Basically, two kinds of metrics are needed to support these decisions. The first is general resource information, like free RAM, disk space or system load. Using this data, it is possible to determine whether the system still has enough free resources, so that additional VMs can be moved to this host. A second metric of importance is defined by the temperature and energy consumption of the system, which is closely related to the PUE. Since the current load and the temperature of a system are closely related, it is possible to correlate these metrics, and to draw conclusions about the energy consumption of the system. Another general metric we identified to be interesting is the current electricity price at each site. Comparing price differences and migration costs, it is possible to evaluate whether energy costs can be reduced by moving VMs from one data center to another. Also, currently available or stored renewable energy can be taken into account for the evaluation.

Given all these data, it is necessary to select the desired strategy regarding the optimization of the energy efficiency. First, it is a good idea to shutdown a server completely if other servers can provide enough free resources to take over its load. Additionally, it is possible to shutdown the servers network switch ports to reduce the energy consumed by the network as mentioned in Section II-C. Besides the network, also a shutdown of other dependencies (e.g., storage resources) can be considered. Basically, there are two options to turn servers on and off. The first option is to control the server using Wake-on-LAN (WOL) if the system was put into ACPI status S3 (Suspend to RAM), S4 (Suspend to disk) or S5 (soft off). Another option is to use IP-based switchable power distribution units (PDU) to switch sockets and attached devices on and off. Using this technique, the BIOS should be configured to automatically boot the system after AC power is restored. Also, entire racks with multiple compute, storage and network equipment could be powered on and off in a controlled way, if an appropriate mechanism exists (and the contained components tolerate the shutdown, e.g., network equipment) to optimize the energy consumptions based on the strategy discussed in this section.

As shown in Figure 4, we introduce a new management component, which has a global view over all servers in the data center. Furthermore, management data from other data centers is collected to get a global knowledge about the resources at every site. The management component collects data from the computes nodes in the data center using REST requests to communicate with the OpenStack API. Based on the collected data, the management component decides when to move VMs by instructing the involved compute nodes to start a live migration process. When this process is able to free enough resources, so that one compute node becomes idle, the management component should take actions to shutdown or hibernate the corresponding compute node to save energy.

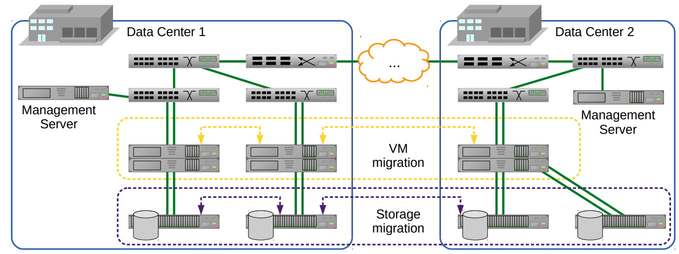


Figure 4. Integration of power management components to enable energy-efficient compute, storage and networking.

B. Energy-efficient Network Connectivity in OpenStack Environments

The complexity of computer networks with respect to energy consumption can be reduced to nodes and links of the network as described in Section II-C. Regarding the energy efficiency of a network in an OpenStack environment, two factors driving the energy consumption can be identified. First and foremost, the energy requirements are defined by the amount of nodes and links. This especially includes power dissipation at each component. Second, the utilization of each node influences its individual energy consumption. The higher the utilization, the more energy is needed for each component. However, as described in Section II-C, current networks are not energy-proportional, so the power consumption of the nodes is not proportional to the utilization of the links. Nonetheless, a sufficient utilization of all links and components leads to increased efficiency. From a theoretical point of view, the network in OpenStack environments builds a graph, with each edge representing a link. By calculating the minimum spanning tree, it is possible to identify the minimum number of links needed to connect all active components. Each link, which is not part of the minimum spanning tree represents a possible candidate to shutdown. However, the problem remains to calculate the preferable spanning tree, considering the energy efficiency and current load of each link, as well as the preferred minimum redundancy, which may differ depending on the specific network segment (i.e., core links connecting the data centers). A promising solution, which considers these requirements has been presented in [21].

To include the metrics of each link in the network a weighted graph can be defined, where the weights of the edges represent the load or utilization of the link, its performance (latency, bandwidth, jitter, failure rate) or in our specific example the energy consumption. By using a graph database (e.g., as part of a network management system), it is possible to model the topology of a network and apply energy consumption metrics to contained nodes and links. Besides classical spanning tree algorithms, as described above, algorithms that allow redundant paths and hence enable fault tolerance and load-balancing, like multiple shortest path trees, can also be used to detect the energy-efficient network topology based on the weights of the links. Network connections of a VM are given by one or multiple paths in the graph. Querying the database, the energy requirements of the network can be evaluated. Also, constraints like fault tolerant links can be defined in the database, as already described in Section II-C. Furthermore, this way the management servers are able to identify redundant links and nodes that can be turned on or off depending on the current utilization of the active links or resilience requirements. Hence, graph databases can be used to support the decision for energy-efficient network connectivity of VMs. Given the dependencies and metrics represented by weights in the graph, components and links can be deactivated or throttled, e.g., during off-peak times, or reactivated based on network utilization.

The management servers can also use the OpenStack network node (using OpenStack Neutron) or an external network management system to support the decisions regarding energy-efficient network connectivity. For example, the network management system or OpenStack Neutron could imply specific topology or performance constraints in the OpenStack environment that need to be considered despite the possibility to minimize the power consumption by deactivating, throttling or suspending network links and nodes.

V. Enhancing the Energy Efficiency of Virtual Machines in Private Clouds using AEQUO

Based on the latin word for equal, we named our prototype AEQUO, as it implements a management component to balance the power requirements in OpenStack environments. The prototype is part of a research project at the University of Applied Sciences Fulda with the purpose of creating a proof of concept to enhance the energy efficiency of cloud environments. In this section, we describe the implementation of our prototype based on the requirements that we defined earlier in Sections III and IV.

A. Implementation of AEQUO

AEQUO is implemented in Python, which integrates well into the testbed, as most of OpenStack's components are written in the same language and offer a Python API. The current implementation consists of a central management component which resides on the controller node. In the current prototype the CPU utilization is the only metric used to determine whether a VM should be migrated or not. Other metrics, such as the current electricity price at a site as mentioned in Section II-B, are also feasible options. AEQUO is getting the data and metrics about the compute nodes and the VMs using REST requests. The REST API is provided by the OpenStack Python API. Before it is possible to retrieve such data, it is required to authenticate against OpenStack. Figure 5 gives a brief overview of the architecture of AEQUO and the REST communication with the Openstack API.

After successful authentication against the OpenStack API, AEQUO starts to get basic data about the compute nodes, such as their name, details about the CPU (i.e, number of cores) and the current status (active or shutdown). The process of acquiring this data happens without any user intervention,

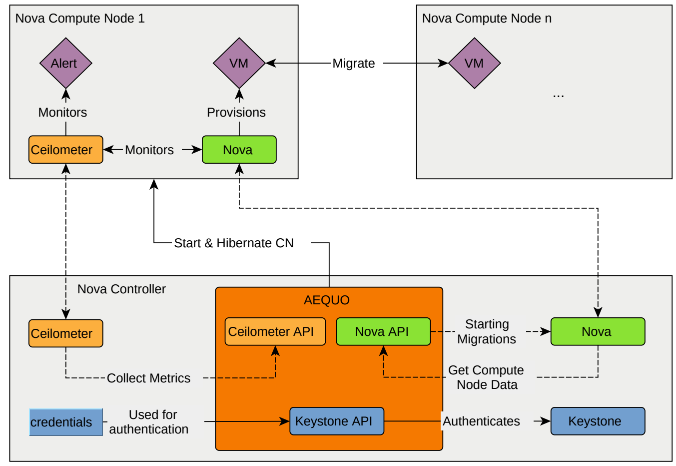


Figure 5. Architecture of AEQUO and integration into OpenStack.

since the data is provided by OpenStack Nova. At this point, AEQUO has an overview over all the active compute nodes. The next step is to acquire basic data about the virtual machines running on these compute nodes. This includes the current state of the machine, the name and the ID which identifies the VM. The prototype only includes VMs in its calculations that are in an active state, hence, VMs that are not shutdown, migrating or in a failed state. This data is also acquired by querying Nova via REST.

After the acquisition of the metadata, the prototype queries the Ceilometer API about the metrics (e.g., CPU utilization) of the compute nodes and the VMs running on them. This process does not include VMs that are already shutdown, migrating or not in an operating state. The prototype also requires a list of compute nodes it should manage. This list contains the nodes' MAC addresses, to start them again if needed via Wake On LAN. It would be possible to automatically collect the MAC addresses of the nodes (though they are not provided by OpenStack Nova), but this would require that all compute nodes are running when AEQUO is started, which is not desirable. For this reason, the prototype requires a list (in form of a file) that is read by AEQUO upon startup containing the compute node names and their corresponding MAC addresses. This also enables the user to define compute nodes which should excluded from the management by AEQUO.

When all the data is acquired, the prototype starts to process it. First and foremost, the prototype detects two states of each compute node's utilization in which the energy consumption can be improved. The first state is the "overloaded state", in which a compute node is considered to be operating near to its maximum amount of physical resources. This critical level is defined by a hard limit. In the current prototype a compute node is considered to be overloaded, if the average CPU utilization was over 90% in the last 2 minutes. If a compute node is in the overloaded state, AEQUO evaluates which of its VMs could be moved. If there is more than one compute node being overloaded, the one with the highest CPU utilization will be selected. Two minutes after a VM was moved from an overloaded compute node, the situation is reevaluated. If the compute node is still overloaded, the next virtual machine will be moved and so on.

In each step, AEQUO individually evaluates where the VM can be migrated, to accomplish an optimal placement with respect to the total power consumption. It is a bad idea to select the compute node with the lowest CPU utilization as the target for the migration, because this compute node might have been prone to be underloaded. Therefore, the compute node with the highest utilization that is still able to handle the additional load of the moved VM without getting overloaded after the migration should be chosen. This is accomplished by looking at the current load of the compute nodes and adding the load of the VM that is to be moved. If the resulting value is lower than 90%, the destination compute node still has enough resources free to accommodate the VM. In case that there are no free resources available and a compute node is prone to be overloaded, AEQUO just prints a warning message. Since the prototype also supports hibernating unused compute nodes, there might be some nodes that could be started to get the necessary resources for the migration.

Despite of overloaded compute nodes, hosts can also be in an "underloaded state". In contrast to the overload state, underload detection is not done by simply looking at the lower limit. Instead, the prototype picks the compute node with the lowest current load and evaluates if the load of this node can be distributed onto the remaining nodes without causing them to be overloaded afterwards. Since there are now probably several VMs to be moved at once, an appropriate result needs to be calculated, before AEQUO takes any actions because there might not be an optimal solution. The problem that needs to be solved is a bin packing problem [49]. Since this is a combinatorial NP-hard problem, the prototype uses a simplified approach based on the first-fit decreasing algorithm. This approach may not provide an optimal solution, but requires $O(n \log n)$ time instead of $O(n^2)$. If a solution was found, it will be applied to the cloud environment, meaning that all the VMs of a underloaded compute node will be moved to a new location. After each migration that the prototype instructs Nova to do, using the corresponding REST API, AEQUO will pause for about 2 seconds. This break is used to give Nova and libvirt enough time to process all the messages being sent to carry out the migration. In other environments, the required timespan for this pause might be different depending on how many VMs a compute node might have on an average and how long it takes to migrate them to another location. After all VMs have been migrated away from the selected compute node, the node can be hibernated, which is currently done by AEQUO using SSH.

All these calculations are scheduled periodically at fixed times. At the beginning all required data will be collected and analyzed. Depending on the results, appropriate actions will be executed and the process starts over again.

B. Optimizing the Energy Efficiency of Virtual Machine Placement and Network Connectivity in OpenStack Environments

As we already mentioned in Sections II-C and IV-B, there are also opportunities to reduce the energy consumption of the network components. Using AEQUO with its capability to monitor and control compute nodes, we currently prepare the infrastructure and graph database to extend our prototype to manage network devices. A possible scenario would be to completely power off a 19-inch rack, including all contained networking equipment like the ToR-Switch (top of rack) as well as the cooling for the rack. Therefore, it is necessary to make AEQUO aware of the components in each rack, and the energy consumption of these parts. This is necessary to support decisions, in which the entire load can be moved from a rack that could be subsequently shut down. At this point, we are evaluating to include asset/facility management or monitoring tools serving as an additional data source for AEQUO.

Another possibility to save energy is to shutdown redundant paths and network devices or links that are only needed at peak times. The devices could be powered off completely by using power distribution units (PDU) as mentioned in Section IV-A. Alternatively, some network devices (e.g., Cisco IOS routers or CatOS switches) have CLI support to power modules or ports up or down. To use these functions, AEQUO needs to be aware of the network structure, to decide what parts of the network can be powered off. As mentioned above, we are currently implementing a graph database as defined in Section IV-B. Instead of shutting down the links completely, network components that support Energy Efficient Ethernet (EEE), as described in Section II-C or techniques that control the power used by individual ports of the switch, could also be integrated, e.g., to throttle the link speed or enter EEE's low power idle mode. As described in Section II-C, the power reduction in this case comes with the drawback of increased latency, which has a negative impact especially on real-time applications. Hence, AEQUO can be used to temporarily turn on EEE and related mechanisms in the networking components when no real-time applications are used (e.g., less VoIP applications or video conferencing traffic during the night). Furthermore, the activation and deactivation of power management mechanisms can also be configured on redundant network paths, as illustrated in Figure 3.

C. Network Device Standby and Power Management

As described in Section II-C, existing network power management solutions could be enhanced to include mechanisms to power off network functions, links and components. The ElasticTree project [21] already referenced the possibility to introduce standby or sleep functions for networking hardware. As such standby power management functions were (and are still) missing in network equipment, [21] suggested to power down idle or underutilized switches completely. From our observations with network equipment from Cisco, HP and Arista, this approach has several drawbacks. For example, the Arista 7050S-52 and 7150S-24 switches (running Arista's Extensible Operating System (EOS) [50]) we used in our testbed in Section VI, take about 5 minutes to boot after being powered on again. Furthermore, powering down the entire switch is currently only possible using external power distribution units (PDUs) as described in Section IV-A. Since it is designed to run continuously, common professional networking equipment does not even have a power switch. As we used the two switches in a multi-chassis link aggregation (MLAG) setup, with each server being connected to both switches, we were able to power down one of the switches, e.g., during off-peak periods, without communication loss in the entire network.

Using external PDUs, the switch is unaware of being powered off, hence we observed minor communication disruptions (i.e., dropping some frames, causing TCP congestion control to reduce the bit rate, spanning tree topology changes due to suspected link flapping). Also, frequent power disruptions might impose risks to the software and hardware state of the switch, though we did not observe such effects in our tests. Furthermore, we simulated the switch power management by using virtualized network operating systems (namely, Arista's Virtual Extensible Operating System (vEOS) [50]) in a VMware vSphere environment. Using this environment, we were able to suspend the VMs running the virtual switches and resume them, to simulate a suspend-to-disk function in the switch. Again, only minor communication disruptions (as described above) were noticeable. Compared to the boot time of our Arista Hardware using EOS, the resume of virtualized EOS (vEOS) took only up to 5 seconds in our tests.

Since Arista is using a Linux Kernel in their products, we investigated further to evaluate the possibilities of a ”suspend to ram” solution. The EOS firmware is formed by a package of a patched Linux Kernel (3.4.43 for EOS 4.14.2F), an initial ramdisk, a root file system (squashfs) and a boot configuration. Arista offers the sources for the firmware and the contained packages on their website [50]. After modifying the sources and adjusting the parameters of our cross-compilation environment, we were able to compile a new kernel containing power management support (i.e., ACPI) and build a custom EOS 4.14.2F firmware. The kernel module for the Application Specific Integrated Circuits (ASICs) used in the 7150S (Intel FM6000 [51]) could not be compiled due to missing sources, hence some modules from the stock firmware were used. Using Arista's API, currently unused ports and their transceivers were successfully switched off (using 10GBase-SRL SFP+ this procedure saves ˜2 Watt/port). Supend-to-disk (ACPI S4) and ”suspend to ram” (ACPI S3) were not unusable, as the BIOS provided with the switch (Arista's Aboot-norcal2-2.0.9) does not fill the necessary ACPI tables. Also, watchdogs and custom patches introduced by Arista limited the power management and shutdown abilities of the switches. However, Arista uses the open source BIOS coreboot in its switches that does support filling the ACPI tables with S3 and S4 capabilities.

The hardware layout used in the 7150S switch consists of a general purpose motherboard (based on AMD Tilapia Fam10 reference design [52]) and a network switch ASIC (Intel FM6000) being connected via PCIe. Patches developed by Arista removed the ACPI capabilities of the coreboot firmware, though the AMD Tilapia Fam10 board supports them [53]. After modifying the supplied sources and adjusting our cross-compilation environment, we successfully compiled a new coreboot firmware for the switch. As the Tilapia board also includes the management network port, we were able to unload the ASIC (Intel FM6000) fpdma kernel module and use PCIe power management. However, significant power savings would require ”suspend to ram” support, which requires further modifications to the firmware (e.g., saving registers in non-volatile memory using AmdS3Save [53]). Another challenge is waking up the switch after a successful suspend. Wake-on-lan on the management network port of the general purpose motherboard is theoretically possible, e.g., combined with separate low-power proxy devices as introduced in [54].

Several papers presented theoretical approaches to implement power management functions in networking hardware (i.e., routers and switches), e.g., also using sleep, standby and rate-adaption techniques [55][56]. Some of them also focused on power management in LAN switches [57][58][59] combined with Energy Efficient Ethernet (EEE) in [60][61]. An interesting combination with the suspend functions we discussed above, might also be the migration of virtualized routers and switches as presented, e.g., in [62]. Energy-aware deployment of routers might also offer advantages to ensure connectivity across distant data centers, as shown in Figure 4.

Other recent related work shows the possibilities of employing power scaling mechanisms in custom built routers, e.g., based on NetFPGA cards [63]. We are currently evaluating to include such sleep and power scaling techniques in our testbed. An option would be to develop custom physical or virtual network switches (e.g., as described in [63]). The other option would be to test the integration of power management mechanisms in existing typical data center networking devices. Upcoming white box switches and open network operating systems (e.g., [64][65]) are offering new possibilities and less restrictions compared to not entirely open platforms like Arista's EOS. White box switches (e.g., [66][67]) also use merchant silicon ASICs for the data plane and typically common x86 architecture for general purpose CPUs and the control plane that include ACPI power management in their reference design. However, as described for the Arista devices above, these features are typically disabled or unused today.

Some network hardware providers already offer network power management frameworks (e.g., HP Adaptive Power Architecture [68], Cisco EnergyWise [5]). For example, Cisco EnergyWise can control EEE and hibernation functions of Cisco Catalyst 2960-X switches [69]. However, these solutions suffer the same problems as described above for our testbed. The switch can be put into hibernation using EnergyWise API or the CLI at the switch, but wake-up can only occur at a specific previously scheduled time or by manually pressing a button on the switch. While this is applicable for small offices or shops, e.g., at night or during non-office hours, traffic patterns of networks in private clouds are hard to predict and hence hibernation cannot easily be scheduled on a regular basis. This also holds true for multipath environments and redundant network devices, as described in Section II-C, as the performance requirements of private cloud networks typically cannot be foreseen in all cases. Therefore, besides the power management functions that are integrated in current networking devices (e.g., CPU frequency scaling, EEE), holistic power management frameworks that are able to toggle the power on temporarily unused or redundant links, custom ASICs or enable sleep or rate-adaption for an entire switch, are an upcoming challenge for continuously increasing bandwidths and the power consumption of today's network infrastructures as described in Section II-C.

Using our AEQUO prototype, as described in Section V-A, we can control the power of temporarily underutilized ports in the multipath network infrastructure shown in Figure 6. Additionally, AEQUO can issue CLI commands to the network switches that deactivate ASIC modules or use ACPI and PCIe power management (Active State Power Management (ASPM)) as presented in this section. Moreover, our AEQUO prototype can be combined with external network management and monitoring platforms (e.g., OpenNMS [70]) or data center infrastructure management (DCIM) solutions. For example, the network management system could inform AEQUO about a planned outage to ensure that all necessary redundant links and components are up, or AEQUO in turn could send information about current placement (e.g., across multiple sites, WAN links) of virtualized networking functions (e.g., routers, firewalls) to the network management system to reflect and monitor the changes in the network topology. Arista EOS and custom virtual network switches also support issuing CLI commands using OpenFlow. Therefore, AEQUO could also inform a central software-defined networking (SDN) controller instead of the network management system about its power management decisions. This way, the network could be dynamically adapted to ensure performance and fault-tolerance requirements within or across multiple data centers while also enhancing the energy efficiency of its links and devices.

VI. Experimental Testbed

In the previous sections, we introduced procedures to automatically migrate virtual machines between data centers. This enables us to consolidate the virtual machines of an organization onto a minimum number of physical servers and to shutdown unneeded components. As a result, we are able to increase the energy efficiency of an organization's actively running IT infrastructure. However, the procedure of optimizing the virtual machine placement introduces load on the infrastructure and causes additional energy consumption. To get an idea of the energy impact of virtual machine migration and energy savings from shutting down physical components, a testbed has been set up at two sites in Germany. We have improved the initial test environment described in [1] with modern components, which are widely used in regular data centers, allowing us to simulate a typical cloud environment with independent compute, storage and networking components.

A simplified overview of the Fulda University site test setup is depicted in Figure 6. At the compute layer, the testbed consists of four identical Dell PowerEdge R620 servers (for reasons of clarity, only two compute nodes are shown in Figure 6), each equipped with two Intel Xeon E5-2650 processors and 256 GB of memory. A unified storage backend was built by utilizing two NetApp E2700 systems with a total of 48 SAS drives, which each compute node is connected to using an independent 16 Gbit/s fibre-channel link. The networking layer was built using the Arista datacenter switches 7050S-52 and 7150S-24, which provide 52 and 24 10-Gigabit-Ethernet ports, respectively. Each compute node is connected to both of the switches using a dedicated 10 Gbit/s fiber link. To setup a cloud environment on the described hardware the OpenStack Icehouse 2014.1.3 release on an Ubuntu 14.04.1 LTS platform was chosen. Two physical servers were used as dedicated compute nodes with Openstack Nova, whereas all other OpenStack components including block storage (Cinder), networking (Neutron), dashboard (Horizon), image (Glance), orchestration (Heat) and telemetry (Ceilometer) services are running on the two remaining servers, which are not shown in Figure 6 as described above. By this, we are able to perform measurements of VM migrations without side-effects introduced by the OpenStack infrastructure.

A primary requirement of our testbed is the feasibility to log detailed measurements of the individual components' power consumption. To meet this requirement, two Raritan PX2-5260R power distribution units (PDU) have been installed, each using an independent electric circuit and providing 12 separately measured power outlets to connect our equipment to. The measurement accuracy of our PDUs was verified by using a professional digital power meter of the type Yokogawa WT333. The discrepancies found were minimal and can be neglected. All components at the compute, storage and networking layers are equipped with redundant power supply units (PSU), which are connected to each of the PDUs. Further, we installed the network management system OpenNMS to continuously poll both PDUs to get the current power consumption of each power outlet. The resulting data gets stored in a round robin database (RRDTool) for further processing and graph generation.

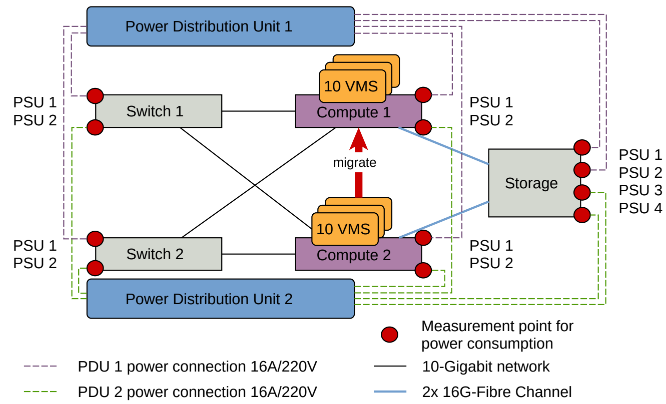


Figure 6. Test environment and network cabling.

The described testbed enables us to perform various test cases (i.e., VM migration, suspending physical components) under reproducible circumstances, as well as detailed measurement of the energy consumption of each component. In addition, another full-featured cloud environment has been set up at the Clausthal University of Technology, providing compute, storage and networking layers in a similar manner. The testbed is running OpenStack Icehouse as well and was connected to University Fulda as an OpenStack region for testing purposes. In the future, this setup will allow us to perform more realistic measurements of virtual machine migration, by taking much more metrics (e.g., WAN latency) into account.

VII. Evaluation

Using the setup described in Section VI, we constructed a test case to get an idea of the impact of virtual machine (VM) migration and network interface suspension on the overall power consumption. The test case consists of multiple phases, which are executed automatically one after another, measuring specific characteristics of the overall power consumption.

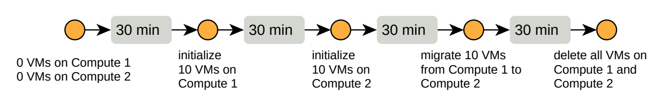


Figure 7. Testplan for measuring different load scenarios.

Figure 7 depicts the procedure of the test case, starting with both compute nodes in idle mode (no VMs running). At an interval of thirty minutes, first ten VMs on compute node 1 and another ten VMs on compute node 2 are spawned. Afterwards, all running VMs are migrated from compute node 2 to compute

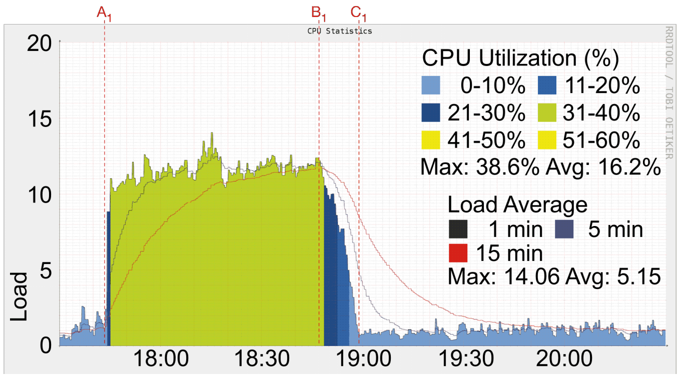


Figure 8. CPU utilization of compute node 1.

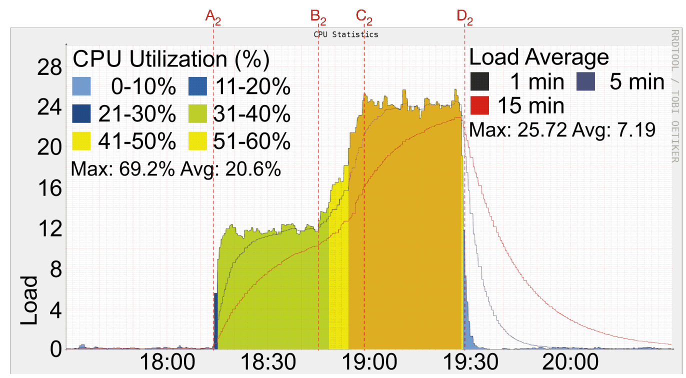


Figure 10. CPU utilization of compute node 2.

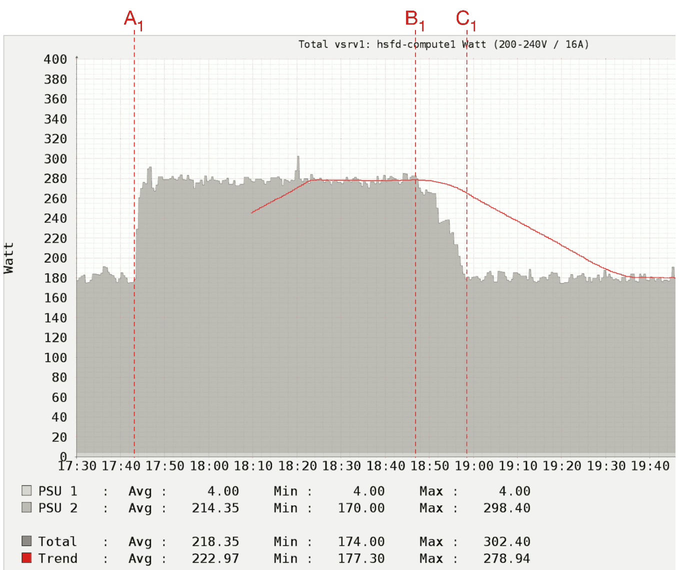


Figure 9. Power consumption of compute node 1.

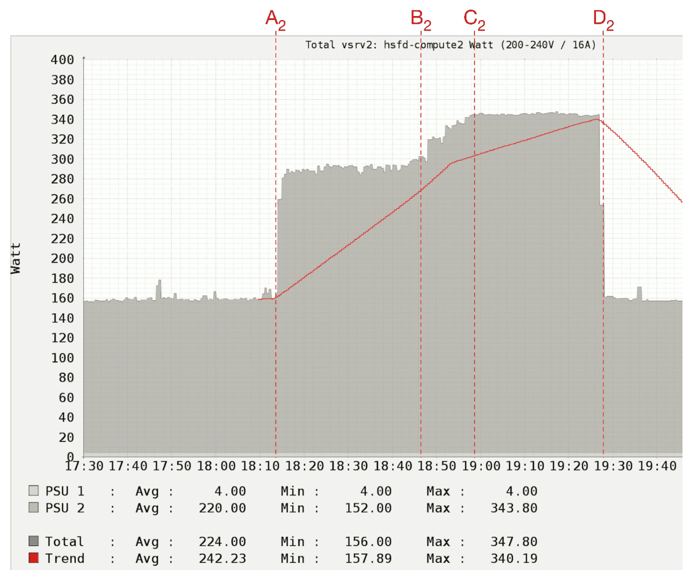


Figure 11. Power consumption of compute node 2.

node 1. Finally, all VMs are removed from both compute nodes. While the test was running, the CPU utilization of the compute nodes, as well as the power consumption of all of the involved components was continuously measured. Figure 6 shows the measurement points for power consumption in our testbed. Further, Figures 8 and 10 reveal the CPU utilization while Figures 9 and 11 show the power consumption for each of the compute nodes. Beside the CPU utilization the metric *Load* is measured to get an idea of the overall system capacity. Basically the *Load* metric is the moving average value of the number of queued processes waiting for execution. A *Load* value of 1.00 states that a system equipped with one CPU is running exactly at its capacity. A single CPU-system running at a *Load* of 2.00 has exactly the same number of processes queued as processes currently running. Of course, the *Load* metric also depends on the number of usable CPUs. Thus, a server equipped with four CPUs will reach the limits of its performance capability at a load value of 4.00.

The different phases of the test are denoted by red dotted lines in Figures 8 to 12. The first step of creating 10 VMs on compute node 1 is visualized in Figures 8 and 9 starting at mark A_1. The second step of creating the same number of VMs on compute node 2 is visible in Figures 10 and 11, starting at mark A_2. All VM instances created on the compute nodes are set up to use the *stress* utility to generate a consistent CPU load. To prevent running into system limits, the *stress* command is configured to generate a maximum of 40% of CPU utilization while keeping the overall system load below a value of 16 (the number of cores available) on each compute node. This leads to a CPU utilization of about 35% at the physical compute nodes, which is represented by different colors of the area in Figures 8 and 10. The effect of adding load to the compute nodes is also clearly visible in the corresponding power consumption graphs (Figures 9 and 11). While each of the compute nodes consumes about 155 Watt at the beginning of our test, the power consumption increases to about 290 Watt after initializing ten VMs (Figures 9 A_1 and 11 A_2). Our next step was the migration of all VMs from compute node 1 to compute node 2. The beginning of the migration is indicated by markers B_1 and B_2 and the completion is depicted by markers C_1 and C_2 in Figures 8 and 10, respectively. As all VMs are now running on compute node 2, the CPU of the node is now utilized by about 70%. After the last interval of thirty minutes, all VMs were shutdown, which is recognizable at mark D_2 in Figures 10 and 11.

By collecting power consumption metrics from all of the components, we are able to trace the weight distribution in the whole setup. This is needed to include the actual cost of the migration (e.g., additional CPU and networking load) in the overall power economization estimation. Figure 12 depicts the overall energy consumption of our testbed, divided into compute, storage and networking components. The total power consumption with ten VMs running on each compute node amounts to about 555 Watt and is shown starting at mark B. The migration of all VMs to compute node 2, which took around 10 minutes is recognizable between mark C and D. It led to an increased CPU utilization of about 75% and a system load of about 25 on compute node 2. There was no measurable impact on the energy consumption of the storage and networking components while the migration was ongoing. However, an interesting effect is visible regarding the total power consumption of the compute components after the migration phase. With 0 VMs on compute node 1 and 20 VMs on compute node 2, the total power consumption is around 525 Watt, which is about 30 Watt less, than in the case where the VMs were spread over both compute nodes. The effect is clearly visible in Figure 12, when comparing the power consumption of the compute components in phase B to C (ten VMs on each compute node) and phases D to E (twenty VMs on compute node 2). Of course, this effect is a result of components being more efficient when operating on higher load, which supports our idea of consolidating VMs on a minimum possible number of physical hosts.

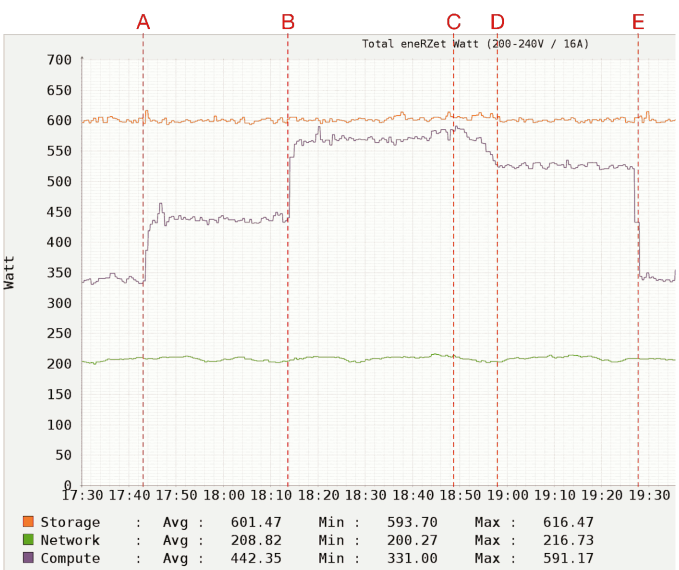


Figure 12. Total power consumption of the testbed setup.

Finally, the possibility of suspending one of the compute nodes was evaluated. For instance, in phase C_1 this would allow us to economize the power consumed by the compute node itself, as well as the power drawn by the networking interfaces it was using to connect to the switches and storage systems. The result of this is not visualized in our figures. However, one can see the effect by subtracting a compute nodes idle power consumption from the total value.

Our measurement of the testbeds energy consumption shed light on possible further enhancements of our prototype and identified non energy-proportional entities in our infrastructure that have a huge potential for energy savings. The AEQUO prototype provides the capabilities to address these promising components and builds the foundation for controlling the OpenStack services based thereupon. Compared the the related work presented in Section II-D, AEQUO offers a novel approach that is able to take different parameters (e.g., utilization, network connectivity requirements, temperature) into account for an energy-efficient placement of virtual resources in existing OpenStack environments. This way, new and upcoming compute and network power management techniques can be leveraged to implement distributed energy-aware private cloud infrastructures. Furthermore, currently available renewable energy resources at different data center sites or different energy prices, as described in Section III, can be used as a parameter for the optimization.

VIII. Conclusion and Future Work

In this article, we presented an overview of our efforts to enhance the energy efficiency in OpenStack-based private cloud environments. Our prototype, called AEQUO, uses OpenStack's standard APIs to monitor the components used in our enterprise cloud testbed. As introduced, AEQUO uses OpenStack Nova and Ceilometer, to acquire the virtual machines (VMs) and hosts running in a private cloud environment together with their current CPU utilization. Furthermore, our prototype is able to control and schedule VM migrations to allow energy efficiency enhancements, i.e., by powering off unused or underloaded recourses. Using wattmeters in external power distribution units (PDU) we measured the power consumption while starting, migrating and shutting down VMs (instances) in our private cloud testbed.

The power measurements revealed as expected, that the power consumption of network and storage components that were used did not change during our tests, in contrast to the compute nodes (servers). This is interesting, because the process of migrating a VM from one host to another requires its virtual hard disk and RAM state to be transfered over the network. Additionally, the hard disk data needs to be persisted on the destination storage system. Therefore, network and storage components we used in our testbed must be classified as non energy-proportional components. Thus, there is potential for further research and optimizations. For example, the network switches in our testbed use merchant silicon instead of proprietary ASICs for the data plane, and standard x86 hardware for the control plane, which supports standard power management mechanisms (e.g., ACPI). Hence, we analyzed, whether we could modify the switch operating system to leverage standard x86 power management functionality. This way, we were able to deactivate unused links and also developed experimental extensions that support to suspend parts of the switch to improve the power management.

Another finding of our test runs was the fact, that there is no linear correlation between the number of VMs running on a compute node and its power consumption. In fact, it seems that the energy efficiency can be enhanced, if the VMs can be executed on a single host, rather than running the same number of VMs on several compute nodes. Of course, this only holds if the the CPU utilization has not already reached the physical maximum. However, in our tests the service quality offered by the virtual machines throughout the tests, which should be impacted by the corresponding consolidation ratio, was not monitored. This leads to the question whether a number of

VMs running on a single compute node can provide the same service quality (besides reliability concerns) compared to the same number of VMs running on several distributed compute nodes. An approach to investigate this aspect would be to use a web server benchmark utility to produce load and gather metrics for the service quality of web servers running in the VMs. Another possible topic for further research that we will focus on is the use of container virtualization instead of type-1 virtualization. In this case, containers are spawned or destroyed on demand without the need to transmit any hard disk or RAM contents. Typically, for example underlying type-1 VMs can provide persistent storage and database services to these containers. Also, the scheduling process of OpenStack could be improved to increase the energy efficiency. For instance, placing machines or affinity groups of dependent machines near their users can reduce network latency and improve user experience. As the prototype that we present in this article contains an extension to the OpenStack scheduler, it could be enhanced to support such an energy-efficient scheduling of new virtual resources (e.g., containers, VMs).

Compared to the related work described in Section II-D, our prototype can be easily integrated in OpenStack environments based on the current Juno release. Further techniques and scheduling algorithms or resource pooling mechanisms can be included in our prototype, thanks to its modularity as described in the implementation section of this article. Currently, our prototype is able to optimize the VM placement and utilization of compute resources in OpenStack environments. Furthermore, we focused on network paths including links and devices connecting the VMs to the network. While energy-efficient networks were also discussed in [21][9][39], our prototype is able to leverage existing and upcoming local power management techniques of compute and networking components (e.g., [9][19]). This way, for example, redundant links in the network can be throttled or even entire devices disabled when network and storage dependencies are integrated into the optimization.

Correspondent scenarios that enhance network and storage power management will be implemented in future testbed scenarios using our prototype. In this article, we already presented several concepts as a starting point to improve network power management. Regarding the storage components, apart from the included standby and power management of the hard disk drives, further investigation is needed. Additionally, our future work includes the evaluation of benefits from different energy prices and lower temperature at multiple sites, e.g., to reduce energy costs for cooling.

As described in this article, we evaluate the use of renewable energy resources for distributed data centers running private/ enterprise or hybrid cloud environments. By measuring the costs for migrations or scheduling of resources in cloud environments, we can evaluate the use of renewable energy at different data center sites or leverage varying electricity prices. As shown in this article, scheduling or migrating virtual resources between data centers in northern, central or southern Germany already has potential for energy efficiency enhancements on the basis of days. We are currently working on a simulation model to evaluate power consumption, migration costs and the use of available renewable energy at distant data centers. Upcoming frameworks for the acquisition of energy consumption metrics in OpenStack (e.g., KWAPI and IPMI extensions) that are currently under development also offer promising possibilities for further extensions to our prototype.

Acknowledgment

The authors would like to thank the Hessen State Ministry of Higher Education, Research and the Arts (HMWK) for partially funding the research presented in this paper within the "Putting Research into Practice" program.

References

[1] K. Spindler, S. Reissmann, and S. Rieger, "Enhancing the energy efficiency in enterprise clouds using compute and network power management functions," in ICIW 2014, The Ninth International Conference on Internet and Web Applications and Services, 2014, pp. 134–139.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST special publication, vol. 800, no. 145, 2011, p. 7.

[3] C. Pape, S. Reissmann, and S. Rieger, "RESTful correlation and consolidation of distributed logging data in cloud environments," in ICIW 2013, The Eighth International Conference on Internet and Web Applications and Services, 2013, pp. 194–199.

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," SIGCOMM Comput. Commun. Rev., vol. 39, no. 1, Dec. 2008, pp. 68–73.

[5] S. S. Sandhu, A. Rawal, P. Kaur, and N. Gupta, "Major components associated with green networking in information communication technology systems," in International Conference on Computing, Communication and Applications (ICCCA). IEEE, 2012, pp. 1–6.

[6] OpenStack, "OpenStack configuration reference - scheduling," 2014, URL: http://docs.openstack.org/trunk/config-reference/content/section_compute-scheduler.html, 2015.05.23.

[7] A. Andreyev, "Introducing data center fabric, the next-generation Facebook data center network," 2014, URL: https://code.facebook.com/posts/360346274145943/introducing-data-center-fabric-the-next-generation-facebook-data-center-network/, 2015.05.23.

[8] J. Baliga, R. Ayre, K. Hinton, W. V. Sorin, and R. S. Tucker, "Energy consumption in optical IP networks," Journal of Lightwave Technology, vol. 27, no. 13, 2009, pp. 2391–2403.

[9] T. Cheocherngngarn, J. H. Andrian, D. Pan, and K. Kengskool, "Power efficiency in energy-aware data center network," in Proceedings of the Mid-South Annual Engineering and Sciences Conference, May 2012.

[10] K. Bilal et al., "A taxonomy and survey on green data center networks," Future Generation Computer Systems, vol. 36, 2014, pp. 189–208.

[11] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," in Proceedings of the 37th Annual International Symposium on Computer Architecture, ser. ISCA '10. ACM, 2010, pp. 338–347.

[12] K. Bilal, S. U. Khan, and A. Y. Zomaya, "Green data center networks: Challenges and opportunities," in Proceedings of the 2013 11th International Conference on Frontiers of Information Technology, ser. FIT '13. IEEE Computer Society, 2013, pp. 229–234.

[13] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan, "A power benchmarking framework for network devices," in Proceedings of the 8th International IFIP-TC 6 Networking Conference, ser. NETWORKING '09. Springer-Verlag, 2009, pp. 795–808.

[14] K. Bilal et al., "A survey on green communications using adaptive link rate," Cluster Computing, vol. 16, no. 3, Sep. 2013, pp. 575–589.

[15] D. Valencic, V. Lebinac, and A. Skendzic, "Developments and current trends in ethernet technology," in 36th International Convention on Information & Communication Technology Electronics & Microelectronics (MIPRO). IEEE, 2013, pp. 431–436.

[16] P. Reviriego et al., "Energy efficiency in 10Gbps Ethernet transceivers: Copper versus fiber," in Optical Fiber Communication (OFC), collocated National Fiber Optic Engineers Conference, 2010 Conference on (OFC/NFOEC). IEEE, 2010, pp. 1–3.

[17] D. Larrabeiti, P. Reviriego, J. A. Hernández, J. A. Maestro, and M. Urueña, "Towards an energy efficient 10 Gb/s optical Ethernet: Performance analysis and viability," Opt. Switch. Netw., vol. 8, no. 3, Jul. 2011, pp. 131–138.

[18] O. Haran, “Applicability of EEE to fiber PHYs,” 2007, URL: http://www.ieee802.org/3/eee_study/public/sep07/haran_1_0907.pdf, 2015.05.23.

[19] K. Christensen et al., “IEEE 802.3az: the road to energy efficient Ethernet,” Communications Magazine, IEEE, vol. 48, no. 11, 2010, pp. 50–56.

[20] Intel, “Energy efficient Ethernet: Technology, application and why you should care,” 2011, URL: https://communities.intel.com/community/wired/blog/2011/05/05/energy-efficient-ethernet-technology-application-and-why-you-should-care, 2015.05.23.

[21] B. Heller et al., “Elastictree: Saving energy in data center networks,” in Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’10. USENIX Association, 2010, p. 17.

[22] A. Beloglazov, “Energy-efficient management of virtual machines in data centers for cloud computing,” Dissertation, Feb. 2013.

[23] A. Beloglazov and R. Buyya, “Energy efficient resource management in virtualized cloud data centers,” in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, ser. CCGRID ’10. IEEE Computer Society, 2010, pp. 826–831.

[24] ——, “Openstack neat: A framework for dynamic consolidation of virtual machines in OpenStack clouds - A blueprint,” Technical Report CLOUDS-TR-2012-4, Cloud Computing and Distributed Systems Laboratory, The University of Melbourne, Tech. Rep., 2012.

[25] A. Song, W. Fan, W. Wang, J. Luo, and Y. Mo, “Multi-objective virtual machine selection for migrating in virtualized data centers,” in Pervasive Computing and the Networked World. Springer, 2013, pp. 426–438.

[26] N. A. Singh and M. Hemalatha, “Reduce energy consumption through virtual machine placement in cloud data centre,” in Mining Intelligence and Knowledge Exploration. Springer, 2013, pp. 466–474.

[27] A. Verma, P. Ahuja, and A. Neogi, “pMapper: Power and migration cost aware application placement in virtualized systems,” in Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, ser. Middleware ’08. Springer-Verlag New York, Inc., 2008, pp. 243–264.

[28] D. Versick and D. Tavangarian, “CAESARA - combined architecture for energy saving by auto-adaptive resource allocation,” in 6. DFN-Forum Kommunikationstechnologien, 2013, p. 31.

[29] X. Wang, X. Liu, L. Fan, and X. Jia, “A decentralized virtual machine migration approach of data centers for cloud computing,” Mathematical Problems in Engineering, vol. 2013, Aug. 2013.

[30] A. Beloglazov and R. Buyya, “Energy efficient allocation of virtual machines in cloud data centers,” in Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, ser. CCGRID ’10. IEEE Computer Society, 2010, pp. 577–578.

[31] T. Carli, S. Henriot, J. Cohen, and J. Tomasik, “A packing problem approach to energy-aware load distribution in clouds,” CoRR, Mar. 2014.

[32] L. Shi, J. Furlong, and R. Wang, “Empirical evaluation of vector bin packing algorithms for energy efficient data centers,” Computers and Communications (ISCC), 2013 IEEE Symposium on, 2013, pp. 9–15.

[33] K. Craig-Wood and P. J. Krause, “Towards the estimation of the energy cost of Internet mediated transactions,” Energy Efficient Computing SIG Technical Report, 2013.

[34] V. C. Coroama, L. M. Hilty, E. Heiri, and F. M. Horn, “The direct energy demand of Internet data flows,” Journal of Industrial Ecology, vol. 17, no. 5, Oct. 2013, pp. 680–688.

[35] D. Hoyer and D. Schröder, “Untersuchung des Potentials von rechenzentrenübergreifendem Lastmanagement zur Reduzierung des Energieverbrauchs in der IKT,” 2009, English: “Study of the potential of using cross-data center load management to reduce energy consumption of ICT”.

[36] T. Huong et al., “ECODANE - reducing energy consumption in data center networks based on traffic engineering,” in 11th Würzburg Workshop on IP (EuroView2011), 2011.

[37] X. Wang, Y. Yao, X. Wang, K. Lu, and Q. Cao, “CARPO: Correlation-aware power optimization in data center networks,” in INFOCOM, 2012 Proceedings IEEE. IEEE, 2012, pp. 1125–1133.

[38] V. Mann, A. Kumar, P. Dutta, and S. Kalyanaraman, “VMFlow: leveraging VM mobility to reduce network power costs in data centers,” in NETWORKING 2011. Springer, 2011, pp. 198–211.

[39] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, “VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers,” Computer Networks, vol. 57, no. 1, 2013, pp. 179–196.

[40] M. A. Adnan and R. Gupta, “Path consolidation for dynamic right-sizing of data center networks,” in Sixth International Conference on Cloud Computing (CLOUD). IEEE, 2013, pp. 581–588.

[41] The Green Grid, “Carbon usage effectiveness (CUE): A green grid data center sustainability metric,” 2010, URL: http://www.thegreengrid.org/~/media/WhitePapers/CarbonUsageEffectivenessWhitePaper20101202.ashx, 2015.05.23.

[42] The Umweltbundesamt, “New ”blue angel” for data centers,” 2015, URL: http://www.umweltbundesamt.de/en/press/pressinformation/new-blue-angel-for-data-centers, 2015.05.23.

[43] B. Valov, “Outlook in the future of german north sea 2020 - power system for offshore wind power,” Wind-Kraft Journal, German Offshore, vol. 5. Auflage, 2008, pp. 38–39.

[44] M. Sterner et al., “Renewable (power to) methane: Storing renewables by linking power and gas grids,” Electricity Storage Association 20th Anniversary Conference, Charlotte, North Carolina, USA, 2010, 2010.05.06.

[45] J. Ringelstein and D. Nestle, “Application of bidirectional energy management interfaces for distribution grid services,” in Electricity Distribution-Part 1, 2009. CIRED 2009. 20th International Conference and Exhibition on. IET, 2009, pp. 1–4.

[46] L.-A. Brischke, M. Hoppe-Kilpper, and A. Tiedemann. Grid integration in Germany. URL: http://www.wwindea.org/technology/ch04/en/4_1_2_2.html, 2015.05.23.

[47] Agentur für Erneuerbare Energien, “www.foederal-erneuerbar.de,” 2015, URL: http://www.foederal-erneuerbar.de/uebersicht/bundeslaender/BW|BY|B|BB|HB|HH|HE|MV|NI|NRW|RLP|SL|SN|ST|SH|TH|D/kategorie/solar/auswahl/183-installierte_leistun/#goto_183, 2015.05.23.

[48] ——, “www.foederal-erneuerbar.de,” 2015, URL: http://www.foederal-erneuerbar.de/uebersicht/bundeslaender/BW|BY|B|BB|HB|HH|HE|MV|NI|NRW|RLP|SL|SN|ST|SH|TH|D/kategorie/wind/auswahl/180-installierte_leistun/#goto_180, 2015.05.23.

[49] M. Sindelar, R. Sitaraman, and P. Shenoy, “Sharing-aware algorithms for virtual machine colocation,” in Proceedings of the 23rd ACM symposium on Parallelism in algorithms and architectures, 2011, pp. 367–378.

[50] Arista, “Arista EOS central,” 2015, URL: https://eos.arista.com/, 2015.05.23.

[51] Intel, “Intel ethernet switch fm5000/fm6000 series,” 2015, URL: http://www.intel.com/content/www/us/en/switch-silicon/ethernet-switch-fm5000-fm6000-series.html, 2015.05.23.

[52] Coreboot, “Coreboot tilapia fam10,” 2015, URL: https://github.com/coreboot/coreboot/tree/master/src/mainboard/amd/tilapia_fam10, 2015.05.23.

[53] ——, “Coreboot AMD S3,” 2015, URL: https://raw.githubusercontent.com/coreboot/coreboot/master/Documentation/AMD-S3.txt, 2015.05.23.

[54] S. Nedevschi et al., “Skilled in the art of being idle: Reducing energy waste in networked systems,” in NSDI, vol. 9, 2009, pp. 381–394.

[55] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, “Reducing network energy consumption via sleeping and rate-adaptation,” in NSDI, vol. 8, 2008, pp. 323–336.

[56] M. Gupta and S. Singh, “Greening of the Internet,” in Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ser. SIGCOMM ’03. ACM, 2003, pp. 19–26.

[57] G. Ananthanarayanan and R. H. Katz, “Greening the switch,” in Proceedings of the 2008 Conference on Power Aware Computing and Systems, ser. HotPower’08. USENIX Association, 2008, p. 7.

[58] M. Gupta and S. Singh, “Using low-power modes for energy conservation in Ethernet LANs,” in INFOCOM 2007. 26th IEEE Interna-

tional Conference on Computer Communications. IEEE, May 2007, pp. 2451–2455.

[59] M. Gupta, S. Grover, and S. Singh, “A feasibility study for power management in LAN switches,” in Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on, Oct 2004, pp. 361–371.

[60] W. Yang, J.-H. Jung, and Y.-C. Kim, “Performance evaluation of energy saving in core router architecture with low power idle for OBS networks,” in Information Networking (ICOIN), 2012 International Conference on, Feb 2012, pp. 318–323.

[61] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, “Reducing the energy consumption of Ethernet with adaptive link rate (ALR),” IEEE Trans. Comput., vol. 57, no. 4, Apr. 2008, pp. 448–461.

[62] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, “Virtual routers on the move: Live router migration as a network-management primitive,” in Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, ser. SIGCOMM ’08. ACM, 2008, pp. 231–242.

[63] N. P. Ngoc et al., “A new power profiling method and power scaling mechanism for energy-aware NetFPGA gigabit router,” Computer Networks, 2014.

[64] Open Networking Lab (ON.LAB), “ONOS - open networking operating system,” 2014, URL: http://onosproject.org/, 2015.05.23.

[65] Cumulus Networks, “Cumulus Linux,” 2015, URL: http://cumulusnetworks.com/, 2015.05.23.

[66] Facebook, “Introducing ”Wedge” and ”FBOSS,” the next steps toward a disaggregated network,” 2014, URL: https://code.facebook.com/posts/681382905244727/introducing-wedge-and-fboss-the-next-steps-toward-a-disaggregated-network/, 2015.05.23.

[67] Open Compute Project, “ONIE hardware status,” 2015, URL: http://www.opencompute.org/wiki/Networking/ONIE#Testing_and_Certification, 2015.05.23.

[68] Hewlett-Packard Development, “Energy efficient networking,” 2011, URL: http://h17007.www1.hp.com/docs/mark/4AA3-3866ENW.pdf, 2015.05.23.

[69] Cisco, “Cisco Catalyst 2960-X platform: The greenest Catalyst switches ever,” 2013, URL: http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/white_paper_c11-728857.html, 2015.05.23.

[70] OpenNMS Group, Inc., “OpenNMS management application platform,” 2015, URL: http://www.opennms.org/, 2015.05.23.

Teamleader App – a Collaborative System Allowing Ad-Hoc Planning Decisions

Sönke Knoch, Pascal Lessel, Melanie Reiplinger, Marcel Köster, Vladimir Pavlov
German Research Center for Artificial Intelligence
Research Department Intelligent User Interfaces
Saarbrücken, Germany
soenke.knoch@dfki.de, pascal.lessel@dfki.de, melanie.reiplinger@dfki.de, marcel.koester@dfki.de, vladimir.pavlov@dfki.de

Leenhard Hörauf, Christoph Speicher
Centre for Mechatronics and Automatisation gGmbH (ZeMA)
Group of Assembly Systems and Automatisation Technology
Saarbrücken, Germany
leenhard.hoerauf@mechatronikzentrum.de, christoph.speicher@mechatronikzentrum.de

Rouven Vierfuß, Torben Joppien
Miele
imperial-Werke oHG
Bünde, Germany
rouven.vierfuss@imperial.de, torben.joppien@imperial.de

Abstract—**This paper proposes a collaborative system that provides decision support for team leaders in an industrial production scenario where staff planning needs immediate adaptations. Requirements were gathered and led to use cases that specify the solution, a mobile application called Teamleader App. A system with an advanced communication protocol was developed, which integrates several sub-systems, such as a supporting domain model and production simulation supporting workers with forecasts of potential decisions. A case study was conducted, which provides first insights into the suitability of the presented solution and revealed further aspects that will improve the system.**

Keywords-Decision support; staff planning; domain model; interaction design; user experience.

I. Introduction

Products highly individualized to fit consumers' needs require flexible production strategies that allow small batch sizes. To deliver manufactured products on time, the person responsible for production has to ensure that the number of planned pieces will be achieved. During production, multiple events occur that may have an impact on the production target. Information technology can help to handle such events. First mobile applications were developed that support people in selected, problem-specific tasks, such as staff planning [1], issue tracking [2] and perfomance measuring [3].

This work is a part of activities towards Industry 4.0 (see [4] and [5]) that bridge the gap between real and virtual worlds and foster the utilization of the potential of the Internet of Things [6]. Industry 4.0 describes the future industrial production as a production of highly individualized products [7] and with an increasing flexibility of production processes. These developments involve a challenge for staff planning engineers who have to answer this flexibility in production with an adaptive staff plan strategy. Increasing quantities of data and complexity make this planning process even harder.

The coordination between planning engineers—in this scenario called team leaders—to find additional qualified workers within the factory consumes time and is a difficult task. To assist team leaders in this process, the concept of a planning support system is suggested. In a defined scenario, a mobile application will support team leaders by visualizing the current worker-to-production-line allocation and by interconnecting the team leaders to coordinate personnel allocation in an efficient way. Therefore, information from several distributed information sources is prepared and presented on a mobile device. Relevant information dependent on the user's role and specific context of use is shown. Human resource allocation can be edited and optimized directly using the user interface.

This article extends previous work (see [1]) concerning the concept for a mobile application to support team leaders in adapting staff plans as a reaction to unforeseen events. Since then, the application was implemented and a first prototype tested by team leaders. The steps during this interaction design [8] process form the subject of the work presented here.

In Section II, an overview of related work is given and the distinctions from the suggested approach are drawn. Section III describes the industrial scenario that is taken as a basis for the considerations made in the following sections. The requirements analysis process including the resulting use cases is outlined in Section IV. It leads to the system and interaction design in Section V. In Section VI, the domain model that, for example, gathers and stores worker profiles and supports the system in its task is developed. Section VII describes how the production is simulated to receive production forecasts. The results of the case study follow in Section VIII. Finally, Section IX contains a discussion of the suggested approach and gives an outlook on future work.

II. Related Work

The suggested system has strong relations to the research field of computer-supported cooperative work (CSCW) [9], also known as groupware. According the definition by Wilson [10] enabling technologies in the Teamleader App case are wireless networks, mobile devices and the merging and aggregation of relevant information. Johansen's Time-Space Matrix [11] divides CSCW systems into four categories. According to this classification, the proposed system is a different time / different place system, processing asynchronous communication. Since systems increased in number and complexity, the classification in the 2x2 matrix became more difficult. Thus, Penichet et al. [12] suggested a more flexible classification method. According to this classification, the Teamleader App is a type C-5 system: C, as it coordinates a production company's internal processes and processes information that enables interaction between team leaders; 5, as it is an asynchronous application that is intended to appear in different spaces (distributed).

Current staff plan software is implemented as a standalone solution or integrated in centralized enterprise resource planning (ERP) or manufacturing execution systems (MES). Time tracking is commonly part of the MES, whereas time management can be part of both MES and ERP [13, 199-212]. Most staff plan or workforce management systems focus on the scheduling and optimal resource allocation task. User interfaces present timetables with a view over a planned period in a very functional way. Some software providers offer mobile applications that allow access to the staff plan systems. These systems lack adaptive context- and role-specific information processing and neglect the collaboration and coordination aspects tackled in this work. MES and ERP systems include high installation, application and maintenance costs. If already implemented, huge efforts and costs are required to adapt such software to a specific use case according to its size and complexity. A problem-specific solution that processes information from such systems to support stakeholders in an efficient way is the target that is focused on in this work.

In research, the project ENgAge4Pro [14] focuses on age-appropriate staff plan. To address the ergonomics topic, physical attributes, such as body weight and height, are considered. The research project EPIK [15] focuses on the optimal allocation of resources to enhance efficiency. A mobile application was developed that supports the worker with context-specific information. The research project KapaflexCy [16] covers short-term production scheduling. A mobile application allows employment requests to be sent to workers. After receiving these requests, the workers coordinate the takeover of the employment themselves. From a hierarchical perspective, this is a bottom-up approach. Unlike the project ENgAge4Pro, the main concern of the staff planning support system is not ergonomics. In contrast to the EPIK project, the system suggested is developed to support team leaders in their planning task. Therefore, more attention is paid to the appropriate presentation of relevant information on the device. Optimized resource allocation will be included in the form of an additional function (not the object of research). Compared to the KapaflexCy project, the mobile application developed here implements the allocation of employment in a top-down way. Nevertheless, worker-related information can be used to provide feedback for workers, e.g., when a mistake during manufacturing has occured [2].

Figure 1. Manufacturing steam ovens at imperial/Miele.

III. Scenario

The production of kitchen appliances in Germany is facing several challenges. Companies require the ability to produce their products under optimum cost and flexibility due to rising variants and a competitive market. Therefore, it is necessary that the manufacturing industry makes efficient use of resources and energy in order to keep the high-cost country of Germany a competitive production location. Manufacturing of steam ovens in imperial/Miele plant floors (see Figure 1) follows the "Miele Value Creation System". Multiple U-shaped production lines for diverse product classes allow for highly flexible handling of varying production programs. Each steam oven is assembled by a single worker in a one-piece-flow setting, which entails high responsibility and a complex work content for all employees.

Once per week, the plant's foremen and team leaders plan the production on the shop floor level, assigning resources and capacities to production orders. Detailed planning is done on a daily basis, considering the production program and availability of workers. In case of unexpected staff shortage or modified production volumes at short notice, the team leaders re-assign available workers to production orders and assembly stations, and also across assembly lines. The process of staff planning is demand-oriented and flexible, and quickly becomes complex and time consuming when trying to meet the demands of multi-variant production scenarios with varying production programs, small lot sizes on multiple lines and customer-individual products. Furthermore, team leaders want to foster a broad skill set in all employees by organizing a rotating assignment of workers to varying tasks while, at the same time, the high quality standards of Miele need to be guaranteed by intense training on each particular product class.

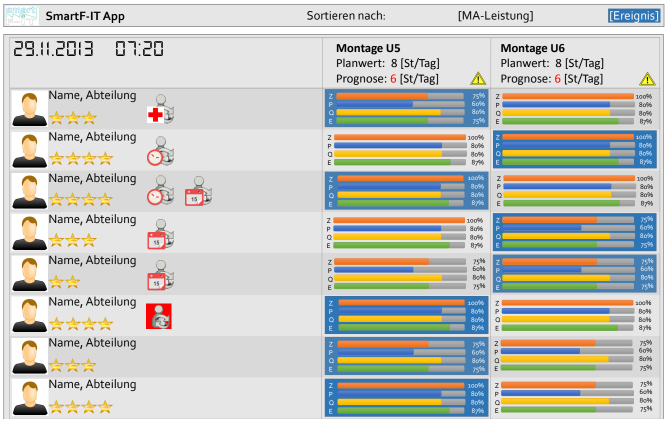


Figure 2. First Mockup of the User Interface.

A great deal of experience is needed in order to make the right decisions. Not all information that is necessary to make the right decision might be modeled and computed. Besides worker profiles that can be automatically generated, soft factors also have to be considered. Thus, an adaptive assistant system that transparently combines data from production orders, human resource management and the plant floor could significantly facilitate and speed up the daily staff planning process in order to support decision-finding.

IV. From Requirements to Use Cases

This section starts with a brief description of the process of interaction design. The most important requirements collected during this process are described and concentrated in five use cases.

A. Process

When the idea of a collaborative tool to support team leaders arose, motivated by the scenario formulated in Section III, the phase of gathering and analyzing requirements [17] started. Discussions between software engineers and experts from the problem domain resulted in a set of use cases. These use cases were formalized according to [18] and described from the user point of view, as suggested by [19] as an essential step in the software engineering process. The most important use cases are described in Section IV-C.

Afterwards, the use cases were discussed and essential data sources to enable the desired application were identified. It turned out that the integration of all these data sources is one of the most challenging parts when implementing such a system in a company. Formats to exchange data were defined and test data was generated. In parallel, first protoypes of the graphical user interface, in the form of paper and PowerPoint prototypes (see Figure 2) as suggested by [20], were designed and improved over several iterations. The resulting prototypes were discussed with the partners at imperial/Miele and the team leaders in the factory to involve users early in the design cycle [21]. The collected feedback flowed into the development of a first prototype that formed the basis for the case study, which is presented in Section VIII.

B. Requirements

In a factory hall up to five team leaders supervise multiple production lines, respectively. To tackle bottlenecks in production, team leaders can request workers from other team leaders (Req. 1). A mobile application could support this process by communicating these requests to all team leaders. Inversely, the same application can provide an overview of the current staff plan, worker profiles and workers' presence/absence (Req. 2). This view has to be exclusive and secured, as staff information is sensitive (Req. 3). The staff plan overview intends to help the team leaders in indicating workers he or she might suggest to change to another team leader when answering a worker request. During this answering process, feedback from the system about the effect of providing a certain worker to a team leader colleague would be helpful (Req. 4). The overview additionally allows team leaders to follow a worker's development, keeping him on track and taking care that he develops knowledge on manufacturing different product variants by switching in defined time intervals between product variants (Req. 5).

To enable these functional requirements, a lot of data has to be gathered and aggregated. Worker and product identification numbers have to be linked to maintain worker profiles. Worker presence/absence has to be taken into account to indicate a bottleneck just in time when it occurs. Numbers about production targets must be made available from the ERP system, allocated to team leaders and production lines. To compute the effect a change in the staff plan might have, information about production lines and product variants has to be modeled. A model that stores skills, relations, and roles is necessary to check access rights to the system (team leaders) and to store and maintain worker profiles.

C. Use Cases

The most important use cases (UC) derived from the requirements analysis are the following five:

UC1 Staff shift (Req. 1)
UC2 Staff presence (Req. 2)
UC3 Team leader authentication (Req. 3)
UC4 Production simulation (Req. 4)
UC5 Staff iteration (Req. 5)

The staff shift (UC1) is the core of the Teamleader App and closely intertwined with the production simulation use case (UC4). To request new personnel for a certain production line, the team leader presses a button. All other team leaders are notified and asked to suggest suitable workers. When a worker is suggested, the production simulation calculates the estimated number of products that will be manufactured if the suggested worker is removed from the suggesting site and added to the requesting site. To deliver fast results to the requesting team leader, the other team leaders are forced to answer within five minutes, otherwise the application is locked. If a team leader has no capacities he can send a "no" or "only in emergency cases" reply instead of suggesting a worker mandatorily. The requesting site receives worker suggestions or rejections together with calculated production values and is able to accept or refuse these suggestions. If accepted, the suggesting team leader is informed and finally requested to confirm the staff shift as his situation might have changed in

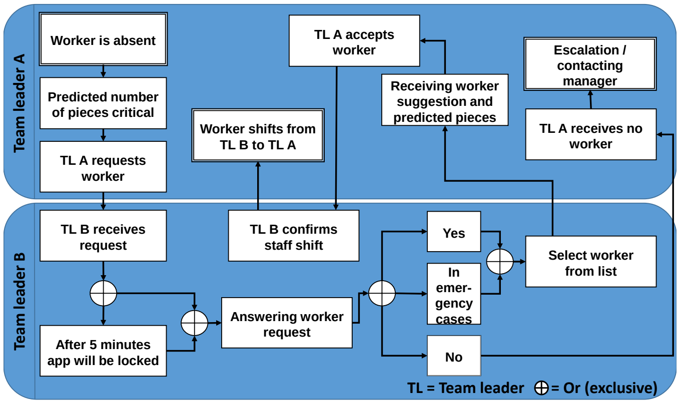


Figure 3. Staff shift use case (UC1) between two team leaders as a process.

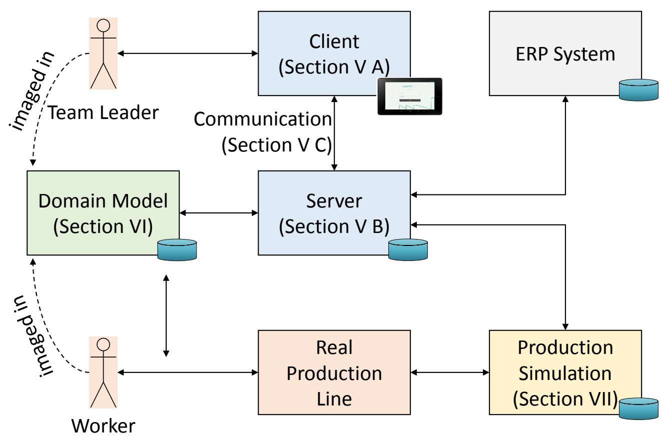


Figure 4. IT-Architecture.

Figure 5. Detailed overview of a team leaders' staff plan.

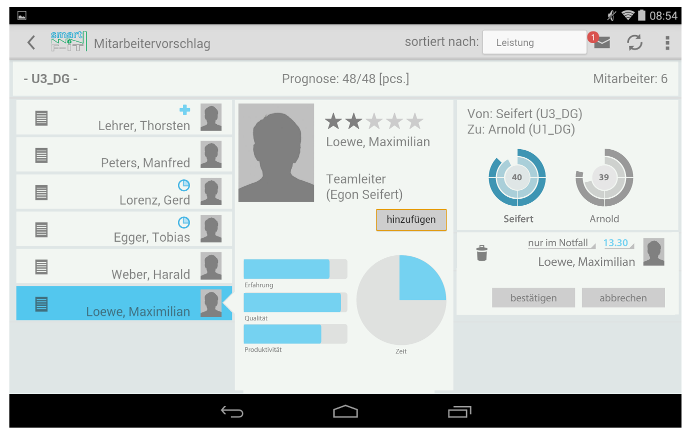


Figure 6. Answering a worker request.

the meantime. Figure 3 shows a staff shifting process between two team leaders.

The staff presence (UC2) enables team leaders to get an up-to-date overview about the status of their workers. The system displays workers that have already arrived at the production line and workers that are absent, e.g., in the case of sickness, unpunctuality, or vacation.

The authentication mechanism (UC3) provides secure access to the information in the system and leads to a view with information relevant to the current user.

A staff iteration (UC5) is desired when a worker has worked for a defined period of time manufacturing a certain product. In this manner, workers stay trained in manufacturing a certain set of products. Therefore, they switch between products in defined time intervals. The system reminds the team leader when a worker should switch to another product.

V. System Design

To tackle the challenges described in Section III and to support the use cases described in Section IV, we developed an application that supports the team leader in adjusting the daily routing when specific events occur. Figure 4 shows the coherences between all components of the suggested system that supports this kind of ad hoc planning. On the *client*-side described in Section V-A a suitable user interface for team leaders that supports all use cases was developed. On the *server*-side described in Section V-B, heterogeneous information sources are prepared and consulted in order to support reasonable decision making on the client-side. The *communication* necessary to support the Teamleader App's core feature (UC1) is described in Section V-C. The server connects to several heterogeneous information sources. An external source is the *ERP System* that delivers information on production planning. The *domain model* reflects all aspects in the application's context, such as workers, team leaders, products, and production lines. It is described in Section VI. The *production simulation* predicts how many products will be manufactured for a given staff plan. It uses its own model of a production line and is described in Section VII.

A. Client-side

1) Screens: The authentication (UC3) forms the entry point to the application. The application loads role-specific profiles for each user. As each team leader manages different manufacturing lines, the respective lines are loaded and currently allocated workers are presented on the home screen shown in Figure 5. This screen has a navigation bar on top, which lets the user switch between detail and overview, sort workers, and contains a link to the message box, a reload function and a context menu. The screen in Figure 5 shows two production lines that are arranged in the two columns on the left. Workers that are allocated to a line have a blue-colored profile in the respective line's column. A profile shows experience, quality, productivity and time values for each worker (in the detail view aggregated on a 5-star scale). The time value, represented by a pie chart, indicates the amount of the recommended time a worker has spent manufacturing a certain product (UC5). The workers are arranged on the right. If a worker is absent (UC2) a symbol signals the reason: a cross indicates sickness, a suitcase vacation, and a clock lateness. In the left column of the home screen in Figure 5, the system shows a deviation from the planned schedule and visualizes a warning by highlighting the discrepancy from the production forecast (UC4) in red and encourages the team leader to act in an ad-hoc manner.

One common reaction to meet the planned quantities at the end of the day is the search for suitable workers. If the team leader decides to request a worker, his colleagues see the screen presented in Figure 6. A red one in the message box symbol at the top of the screen indicates an important message from a requesting team leader. If the receiving team leader switches to the message box and decides to suggest a worker, he will see the screen in Figure 6. On the left of the screen a list with all workers is shown. The middle of the screen shows the profile of the selected worker. A button allows adding an available (not absent) worker to the suggestion message. If that happens, the production is simulated under the new condition that the selected worker will not work on his current line and instead work on the line requested by the other team leader. The charts on the right of the screen show two values: the inner and outer circle represent the number of manufactured products before and after editing the staff plan respectively. The left chart provides numbers for the suggesting team leader, the right chart for the receiving team leader. This visualization is intended to support the team leader in making his decision. Different workers lead to different predictions. The list of suggested workers can be extended to contain more workers. If the suggestion is complete, the teamleader can send his response by pressing a button. When his colleague accepts his suggestion, the worker can shift between the production lines (UC1).

2) Worker profiles: The detail view shown in Figure 5 shows the qualifications of workers. It is possible to switch between detail view and overview. The latter shows a five star ranking for each worker. To compute the qualification of a worker (w) for a respective manufacturing line (l) and product (p), three parameters were defined:

$experience(w, l, p)$
: Total time spent on this line by a worker.

$quality(w, l, p)$
: Defective pieces per shift by a worker.

$productivity(w, l, p)$
: Pieces per shift by a worker.

These parameters allow the team leader a rough estimation of the worker's skills. The ranking function $rank(w, l, p)$ forms a weighted aggregation of these parameters and represents the skill level of each worker on a scale between 0 and 5. These weights are dynamic and adapted to the specific use case.

3) Realization: An eight inch tablet was identified to be most suitable for the daily deployment in a factory environment. It is small enough to fit in a team leader's pocket and offers enough space on the screen to present relevant information.

The implementation is based on a model-view-controller (MVC) framework. The separation into view and controller allows a fast integration of new user interfaces and the reuse of components. Both view and controller access the model, which holds the dynamic data that is presented in the accessing view. In a particular case there exists a bidirectional data binding mechanism. A controller is aware of changes that are made by the user in a view. The view is automatically updated if changes on the model-side occur. In the Teamleader App these bidirectional data bindings are important for the dynamic updates of the user profiles (experience, quality, and productivity visualizations), which change continuously during a working day.

Furthermore, the implementation is separated into different modules. These modules can be included or excluded according to the the specific requirements of an individual factory. This kind of flexibility allows the generation of lightweight and full-featured versions of the Teamleader App, which adapts itself in this manner to its environment and context of use. The content on the application site can be used in a multilingual environment. An integrated translation engine allows one to dynamically change the language of the Teamleader App. This feature enables the company to introduce new languages by defining translation pairs of vocabulary.

B. Server-side

The information that is necessary to support the planning process of a team leader as described in Section V-A is located at three different points that are sketched in Figure 4. In the present case, the information system can be described as a multi-computer, partitioned, distributed, shared-nothing system. Thus, a suitable strategy to integrate the information has to be selected. To preserve the autonomy of the sources, a virtual integration strategy was selected, which leaves the data at the sources. This kind of on-demand integration in a decentralized manner enables us to keep the system design easy to extend and to transfer data only when needed from solely relevant sources. A mediator-based approach [22] was chosen to realize the virtual integration system. The mediator provides an interface implemented as a Web service and communicates with the application. It is the responsibility of the mediator to provide a structural and semantic data integration. Wrappers are implemented for each information source to overcome the heterogeneity on the data level and to enable the data flow between mediator and sources.

Focusing on the data sources, the ERP System is the only existing system that is already available in a common factory. An ERP system in the considered scenario provides—in collaboration or separately—access to time tracking and production planning data. The domain model encodes the workers' skill matrix and working history. It is described in detail in Section VI-B. The production simulation described in Section VI-A contains a model of production lines containing information about process steps and involved manufacturing equipment and allows estimations about quantities that can be achieved when a specific worker is rescheduled on that line. In combination with processing times, it could also be used to calculate optimal resource allocations with algorithms from the operations research field. The factory worker model constitutes new input data that allows new forms of optimization based on the user model and corresponding profiles.

The server was implemented using the inversion of control design concept and dependency injection. This technique allows replacing parts of the server without writing any gluing code lines through reflection. An extension or adaption of functionalities benefits strongly from this approach. The server's authentication process maps every client instance request to the client itself. This approach ensures that only authorized clients can access a specific data slice.

C. Client-Server Communication

The actual process of transferring workers between team leaders involves several steps and states. Figure 7 shows the high-level connection between a team leader *TL* 1 and his colleagues *TL* 2 to *TL* $n-1$. We have chosen a client-server architecture over a peer-to-peer one for two main reasons. First, a lot of different aggregation procedures (like worker-profile based information) require information from a central big-data factory storage that is influenced by many machines and sensors inside the factory. Second, communication to another team leader might be impossible for several seconds due to an insufficient network connection or a crash of a client device. In such situations a communication proxy (like a server) can handle those special cases by realizing a separation of concerns.

Figure 8 shows the possible internal states and transitions of a single worker transfer. The solid black lines, on the one hand, describe possible actions that can be performed by the team leader who initiated the request. The dotted lines, on the other hand, indicate actions that can be triggered by all other team leaders *TL* 2 to *TL* $n-1$.

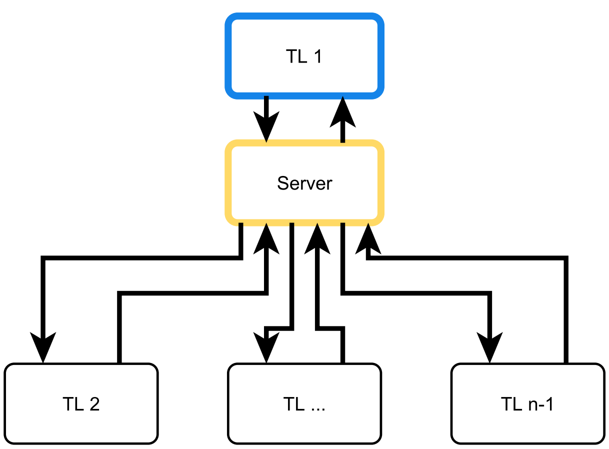


Figure 7. High-level team leader (TL) and server connection

As soon as *TL* 1 initiates a worker transfer by requesting new workers in the client application, the request is sent to the server and the actual transfer process starts. The initialization on the server side causes the newly created request to switch its state from *Idle* to *Requested* (see Figure 8). Information about the request (the production line, the desired skills, etc.) is now broadcasted to the other team leaders. Each team leader can now decide on his or her own to offer possible workers for a transfer or to decline the request. Note that it is not possible to offer the same worker twice, as long as the request, containing the worker is not in the *Closed* state.

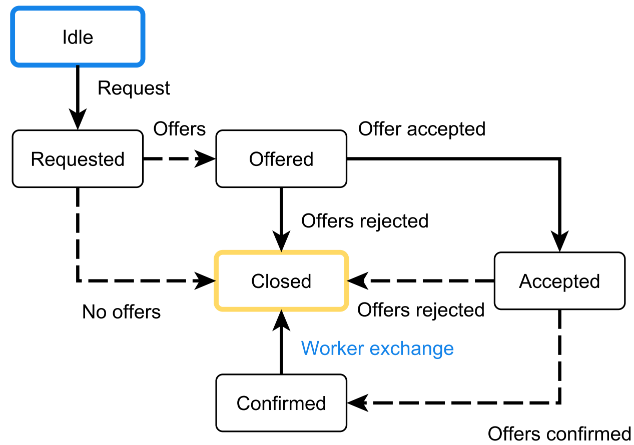


Figure 8. States of a single worker transfer

If no offered workers can be received, the global request will switch to the *Closed* state and *TL* 1 will be notified that no workers are currently available for a transfer. If at least one other team leader offers some workers, the state will switch to *Offered*. In this case, *TL* 1 can review every received offer and can decide whether to accept the proposed workers or not. Similar to the previous steps, a rejection of all proposals switches the whole request to *Closed*. Accepting at least one proposal will trigger a state change to *Accepted*.

Once an offer is accepted, the responsible team leader is notified and can review the proposed workers once again. This additional review phase allows the team leader to reevaluate his or her current situation, which could have changed since the workers were offered. If the transfer is approved, the overall state will be set to *Confirmed* and the proposed workers will be transfered to *TL* 1, since the global request was already successful. Other available offers, which are not confirmed yet, can still be confirmed or rejected afterwards. However, as soon as all responses from all team leaders are available, the global request will switch to *Closed* and the request can be removed from the server.

A frequent exchange of information between clients requires a stable, fast, and reliable transfer of data. To tackle this key issue, a channel-based communication protocol from Section V-C is used. Therefore, incoming and outgoing channels were implemented to support the staff shift procedure. The outgoing channels are divided into broadcast and individual message channels. A client instance subscribes to a channel and will be notified by the server in a given situation. This kind of communication was realized using the WebSocket protocol [23] that provides full-duplex communication channels. The internal information-exchange format uses JSON as intermediate data representation.

VI. Domain Model and Database

In order to directly link the Teamleader App to the actual production processes, namely the concrete current situation in the factory, we integrated a domain model as part of the application's backend. This model consists of a graph, representing a semantic network that flexibly interconnects information from different areas of the plant floor, on different layers of granularity. The network combines a topology of manufactured products with a model of the factory floor and the staff profiles, and is used as a shared domain model by different SmartF-IT applications (such as [1] and [2]). Apart from the product hierarchy, it represents the structure of the industrial facilities, consisting of production lines and their work stations. A third type of data contained in the model is the employee model, providing the application with detailed structural and individual information on the factory staff. In order to make these representations of products, lines and employees a realtime model of the plant floor, several pieces of up-to-date information have to be fed into the domain model. Team leaders and workers are connected to their respective responsibilities and work places. Planning figures, taken from the ERP system, link manufacturing lines to the product hierarchy, enabling the Teamleader App to check on production targets and trigger the simulation of goal achievement w.r.t. the respective staff configuration. Employees are interconnected via hierarchical relationships that depict their areas of responsibility.

A. Product Hierarchy and Facilities

Our approach of representing the domain as a semi-structured data network (cf. Section VI-C) allows for flexible modeling of products and plant floor facilities, customized for the needs of individual enterprises. Depending on the enterprise's requirements of production planning w.r.t. flexibility, minuteness, lot size and breadth of product range, the product hierarchy can be specified to an arbitrary level of detail, starting from product groups or categories (e.g., steamer, cooking chamber), ranging over devices (steamer, combi-steamer) and device types down to a fine-grained distinction between device variants, e.g., country- and market-specific versions of products.

The granularity of modeling the plant floor facilities depends on a factory's individual philosophy and implementation of the production process, and of course on the respective product's inherent properties. If the production consists mainly of one-piece-flow processes with most of the assembly taking place on the same spot, and with the help of a material shuttle, the structural facility model might consist merely of floor areas or work places. In cases of conveyer belt production with clocked processes, or when a more fine-grained modeling of responsibilities and production planning is required, we need to represent individual production lines, work stations, and maybe even process steps (the single operations or work steps, executed automatically or by employees), which are in turn connected to the product hierarchy, namely, to those product variants that require these steps as part of their production schedule.

B. Model of a Factory Worker

The information about workers' skills and experience needed by the Teamleader App in order to generate useful recommendations is taken from the part of the domain model that represents the factory staff. This employee model represents semantic relationships between the horizontal and vertical roles of an employee in the factory (i.e., his tasks, work content, and his position within the staff hierarchy) on the one hand, and his skills (e.g., work experience) and individual requirements on the other hand. Examples for the latter are an employee's handedness, language skills, allergies to specific materials, or an inability to lift heavy weights or to distinguish colors. A small excerpt of the worker model is visualized in Figure 9. The qualification of workers is encoded within the *History* nodes of a semantic network between products, assembly lines, and employees. The application can, for instance, query the model for workers that are experienced assemblers of product p at assembly line l, and rank them using weighted aggregation as described in Section V-A. The depicted subgraph shows how information about a specific worker, *Williams*, is encoded in the model. *Williams* himself is an assembly worker (shown by the *hasRole* relationship), whose native language is English and who currently works at work slot *U6_S05*, which belongs to the slot group *U6*. *Williams* is associated to the product hierarchy indirectly: his work place links via the *produces* relation to the product group *DGC* (combi steam oven), that is currently being produced at *U6*. In order to simulate production processes, the Teamleader App will query, e.g., *Williams*'s working experience w.r.t. the production of *DGC* (via the *hasExperience* relationship).

Of course, such working experience has to be fed into the model before it can be provided to applications such as the Teamleader App. An automatic logging of production data is most convenient for realtime model updates, and many modern factories already implement the foundations of such logging in the form of sensors that are part of the production lines and that monitor, e.g., the execution of single process steps. In enterprises that have no automated logging of process steps, the staff's working experience can be updated, e.g., on a daily basis, using the production data from the preceding workday. Similar to the degrees of freedom in modeling product hierarchies and facilities (cf. Section VI-A), the recording of data can vary in its level of granularity. Simple counting of assembled pieces can be refined to an informed logging about the number of correctly vs. incorrectly assembled devices, with results from the test rig incorporated in the data recording, or even to logging of individual work steps.

Data logging is partially decoupled from an analysis of the recorded data, because evaluation can aggregate over various levels of logging. The assessment of a worker's qualification is done on aggregates of the original logs, e.g., by summing over the number of devices of a specific type, assembled by this worker within a certain interval, or by computing the ratio of correctly executed work steps across devices. This is an important factor in a scenario where not all desired use cases of evaluation, i.e., modes of aggregation, are known at the time of database modeling, or even at the time of data recording. Different applications will have different interests w.r.t. the modeled and recorded data, and therefore compute individual aggregation functions on the graph.

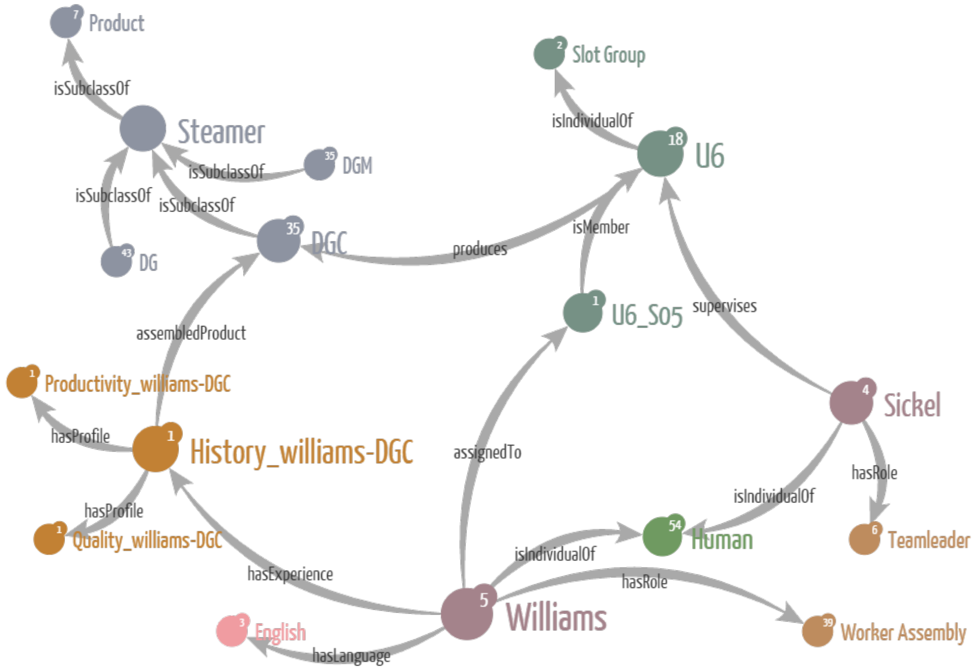


Figure 9. Excerpt of the domain model. Node colors denote the semantic groups of information encoded in the graph (blue: product hierarchy, dark green: plant floor facilities, orange: worker experience as part of the employee model).

In the imperial/Miele scenario, the Teamleader App presents worker experience on the level of product types or variants, meaning that a relatively coarse data logging in the form of daily updated production numbers is sufficient. However, in order to ensure a fair and realistic assessment of worker qualification, we plan to realize a feedback mechanism using signals from the repair stations, in order to integrate the knowledge about whether a device that failed testing was incorrectly manufactured due to a mistake made by the worker, or rather due to other factors, e.g., material deficiency or tool abrasion. This goal entails another prerequisite, namely that each single device that was produced relates back to the workers involved in its production process. In modern factories, this might be realized by barcode stickers or RFID chips attached to the product, possibly even in the form of a digital product memory [24].

C. Implementation of the Domain Model as Database Backend

Our domain model is implemented using the open-source, Java-based graph database Neo4j [25]. Querying of the model from within the application is realized by accessing Neo4j's RESTful interface. The model's contents are derived from a domain ontology (built using the Web Ontology Language OWL, [26]) by automatically mapping the ontology's concepts and relations into the Neo4j graph database format (T-Box). Once created, the graph can be populated and updated (A-Box) at runtime with dynamically-changing data like a worker's history, retrieved by logging of assembly operations.

There are several reasons for choosing a graph database over conventional storage formats like, e.g., SQL databases. Regarding performance for path operations in highly-connected data such as our domain model, relational databases quickly become overburdened by queries of increasing complexity due to joins and index lookups; whereas in graph databases, which use index-free adjacency in traversing from node to node, query latency is relatively independent of the database size and the number of connections (see [27], chapter 2). Note that denormalization for relational databases is not an alternative here, since the data model is not tailored exclusively to the needs of the staff planning app, but is instead meant to provide a flexible, multi-purpose source of semantic information, with diverse applications reading from and writing to the model. This implies that a) we cannot anticipate what relations will be queried most frequently at runtime, and b) we need to prepare for easy model update (e.g., based on sensor data), in order to keep the graph a realtime model of the staff data. Consequently, there is no use in optimizing read access for specific relations at the cost of slower write access.

The most crucial benefit of graph databases in the context of Industry 4.0 is that they allow for an explicit, intuitive, and easily-expandable modeling of the complex semantic dependencies that exist in modern factories, and that the semantics of such graphs can easily be understood even by users unfamiliar with conventional modeling languages like the unified modeling language (UML).

VII. Production Simulation

When scheduling workers for different assembly lines, a team leader is supplied with various information and key figures via the APP. Part of this information are the expected quantity forecasts considering planned workers, production program and assembly times of individual product variants. For this forecast, a simulation is run in the background of the Teamleader App, in which a defined part of the value stream of the steam oven assembly is mapped. The dynamic simulation model, which is prepared with the software Plant Simulation [28], is connected with the Teamleader App via database and is started by a hypertext transfer protocol (HTTP). The data exchange between the simulation model and the database occurs via an ODBC module. In the database, the Teamleader App deposits information such as worker assignment to the lines, their qualifications and production program as parameters for the model initialization.

If the HTTP is called in the Teamleader App, the simulation model will be started. This happens during the rescheduling of the worker to the assembly lines. Thereby, among other things, the new worker allocation is deposited in the database and the model is filled with these parameters. Afterwards, the simulation occurs, taking nearly three seconds for the simulation of an entire shift (7.5 hours). At the end of the simulation, the results of the simulation are stored in a database. The Teamleader App recognizes the new result, reads it, and visualizes it in the front end.

The value stream includes the individual areas of assembly and test stands at the end of the assembly line. In the factory, two organizational forms of assembly occur, which are mapped in the simulation model. On the one hand, steam ovens are assembled in one-piece-flow, on the other hand, fixed position assembly is used at another assembly area (Figure 10 shows an extract of the simulation model). During one-piece-flow, the worker moves with the product to be assembled along different assembly stations. At these stations, the material is provided and the assembly process steps can be performed. Whereas in the areas of the fixed position assembly, the whole assembly process is executed at one station except for the functional test.

The one-piece-flow lines are displayed in the simulation model as assembly rows and consist of five stations and workplaces, which are spatially (edges) and temporally (intermittent, asynchronous) interlinked. In fixed position assembly, separate stations and workplaces are modeled. In both cases, the assembly happens according to the object principle. Figure 10 represents an extract of each of the two assembly areas of the simulation. Each assembly station consists of a station and the workplaces for the worker. In the case of one-piece-flow the worker will move from station to station, following the product until it is fully assembled. For each station the process times (depending on product variants) are stored in a table. A source will provide the assembly station or assembly lines with various products (orders), which are listed in the production program table.

As in reality, the test stands are positioned at the end of the assembly lines, whereas an assembly line is followed by one test stand. The test stands can be fed with complete assembled products by every line. Depending on the inspection scope, assembled products are outsourced to adjacent test stands, which is also considered in the model. Since long distances need to be walked to faraway test stands, which has an impact on the produced quantity, the workers are sent with their products to the adjacent test stands.

The qualification of the worker also has an impact on the quantity. In the model this aspect is considered, due to the factors assembly area, one-piece-flow, fixed position assembly and inspection. This is also reflected by the deposited worker model.

In the first version of the simulation model, no optimization was performed regarding the worker scheduling. The model

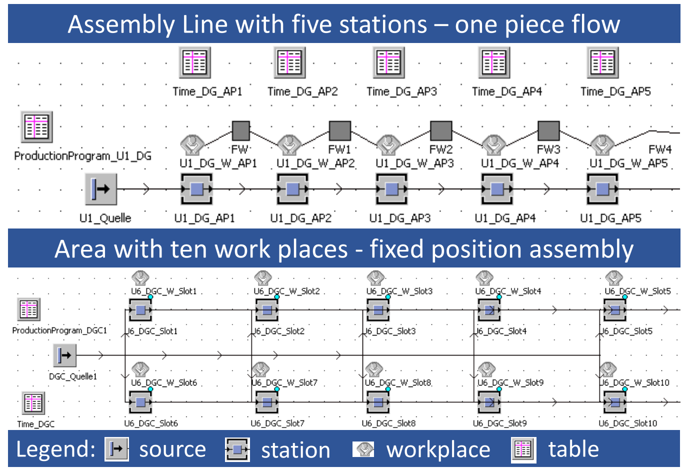


Figure 10. Model extractions representing one-piece-flow and fixed position assembly.

shows possible consequences of a team leader's decision and supports him regarding his selection. The team leader has the final say because many soft framework conditions need to be considered, such as the education of new workers or the performance of a team, which are hard to simulate.

VIII. Case Study

To receive further insights into the requirements and needs of team leaders, we decided to conduct a user study with the additional goal in mind to evaluate the usability of the current prototype (a typical step in the user-centered design process [29]). Specifically, we wanted to confirm the following hypotheses:

- **H1** Team leaders will in general be able to use the current prototype without instructions.
- **H2** Team leaders will appreciate the functions offered and can imagine using them in their daily work.

We also expected the team leaders to provide us with more ideas for functions that could ease their work further.

A. *Method*

The study was conducted during the working hours at the work place of the team leaders. We conducted a single session with each participant and closed the study with a group discussion session in which their supervisor, who had also a general notion of how the app works, was also present. Every session took approximately 30 minutes. The mobile application was shown on an 8-inch tablet.

In respect to the goals of the user study and the hypotheses to test, we decided to let the team leaders explore the application on their own without any explanation beforehand. The participants were encouraged to think aloud while using the system, as suggested by [30] and [31], to reveal positive and negative design decisions and observe their interactions. To ensure that every team leader would be exposed to all implemented features, we guided them to certain functions, if observation indicated that they would fail to find them otherwise. As stated before, some of the team leaders were involved in the earlier requirements analysis and thus might have remembered which functions the system should offer in principle, but no team leader saw the current prototype state in advance. Prior to any interaction, the team leaders answered a pre-session questionnaire collecting general information (e.g., how much experience they have with tablets) and questions about problems in their daily work that could be overcome with technical aids. After the exploratory interaction phase a System Usability Scale (SUS) [32] was provided as well as a post-session questionnaire with quantitative questions (e.g., "I think I can ease my daily work with the team leader app") to be answered on 5-point Likert scales and qualitative questions (e.g., "Which of the features seen needs improvement?").

The group discussion took place three hours after the first single session and focused mainly on the impression the participants had of the application. We thus followed an unstructured interview technique.

B. Results

The study was conducted with five team leaders. Even though the number of participants looks small, we refer the reader to the work of Nielsen [33] in which three to five participants were reported to be sufficient to find nearly all usability problems in a system. The average age of the participants was 36 years (SD=6.78, Mdn=36) and they all reported having intermediate experience (M=3, SD=0, Mdn=3) with computers. Each of them owns a smartphone (reported experience level: M=3.6, SD=0.55, Mdn=4) while only two own a tablet (reported experience level: M=1.8, SD=0.84, Mdn=2). In general, the team leaders reported to be open to technical innovations (M=3.8, SD=0.45, Mdn=4).

1) Potential Technical Improvements: The team leaders are convinced that technical aids can support them in their daily work (M=4.2, SD=0.84, Mdn=4). We asked them to state up to three aspects (without providing answering options) that could be supported and how they could imagine being supported by a system. Interestingly, three of the five participants stated that currently communication seems to be problematic and assistance applications for this might be helpful. Twice an improved access to specific data (e.g., which malfunctions are currently active, or a direct overview of how many products were created on a given day and tested without error) was mentioned. One team leader stated that better mobility of mobile applications would also be beneficial, as access to specific subsystems is currently only available from stationary work stations.

2) Perception of Features and App in General: Concerning the quantitative and qualitative questions in the post-session questionnaire, we learned that the team leaders think that the team leader app is a reasonable innovation (M=4.6, SD=0.55, Mdn=5) and that it eases their work (M=4.8, SD=0.45, Mdn=5). This serves as evidence for **H2**. We asked them (again, without providing any options to select from), which features were perceived as most useful. Table I shows an overview of them. Concerning the results reported so far, the mentioned features are easily explainable. The capability to exchange an employee and the option to receive an overview of workers at assembly lines together with system-derived values helps to make expert knowledge (which every team leader has for his own assembly line) available to others and thus eases the communication among the team leaders. The prediction system, on the other hand, also seems useful as it provides a simple-to-understand number to find out what consequences the exchange will have. This again serves as a formalization of expert knowledge. These aspects were also confirmed in the group discussion.

TABLE I. OVERVIEW OF PERCEIVED MOST HELPFUL FEATURES

Feature	Times mentioned
Exchange of an employee	4
View of worker attributes (experience, quality, productivity)	3
Prediction system	3
Overview of workers at an assembly line	3

3) Usability: The observation of the participants has shown that the participants in general were able to interact with the application without instructions. The core problem observed though was that they were not so sure which areas were clickable (and thus lead to further information) without trying it. Even though this testing helped the team leaders to explore the functionality of the app, we are reluctant to accept this as evidence for **H2** and will test this specifically in the next iteration of our user-centered design cycle. For this, we will also fix the usability problems revealed in this study (by observation and discussion, cf. Table II). The System Usability Score (SUS) [32] supports this further as an average score of 62 (min=50, max=70) indicates issues (as the maximal achievable score is 100). Following the work of Lewis and Sauro [34], we can distinguish between the usability part and the learnability part of the SUS; we still see these issues (usability score of 60.63) but the learnability is slightly better (67.5). The quantitative questions concerning the usability indicated that team leaders think that the usage of an 8 inch tablet is not the worst possibility (M=1.4, SD=0.89, M=1), but also not the best option (M=3.9, SD=1.1, Mdn=3). Also when actively asked, the navigation was perceived as improvable (M=3.6, SD=1.14, Mdn=4) and the graphics seem not to be completely understandable (M=2.8, SD=1.48, Mdn=3). The text parts of the app seamed to be better understandable (M=4.2 SD=1.1, Mdn=5) and the selection of color could be improved (M=3.2, SD=1.3, Mdn=3).

TABLE II. OVERVIEW OF FOUND USABILITY PROBLEMS

Usability issues
• Unclear navigation in terms of which areas are clickable and lead to further information
• The association between a worker and his corresponding line (when a team leader supervises more than one assembly line) is unclear
• Buttons and font sizes are sometimes too small
• The assembly line overview is not completely clear at a first glance (prediction of assembly line outcome was interpreted as prediction of worker performance, which is given only indirectly)
• More explanations on the way the application works are needed (e.g., how stars are derived)
• The exchange of an employee view was not intuitive enough. The different statistics were unclear and the team leaders had the feeling that to make a good decision they need further information that is not yet accessible directly in this view. Additionally, more information on the exchange status should be integrated, as the team leaders were not sure what happens after they have sent a request.
• The arrival of in-app e-mails should be made more obvious.

4) Requested Features: Several features could be elicited that could potentially ease the work of team leaders and would in consequence further improve the perception of the Teamleader App.

- It should also be possible to rate specific workers, to establish a second measure besides the system-derived measurement. In consequence, the team leaders also want to make notes on the different workers.
- More information about the workers should be accessible (e.g., how many products they have built).
- The application should offer the option to exchange workers only for a specific amount of time.
- The application should directly integrate other different sub-systems (e.g., ERP systems).
- There should be more options in the exchange; i.e., it should be possible to directly request specific workers (because experienced team leaders have certain knowledge about many workers) or request workers that fulfill other specific attributes (e.g., can drive a specific vehicle).
- Team leaders want to see the state of assembly lines of other team leaders. Here, they do not want to see every detail, but rather an overview of the line.

As these features focus on making expert knowledge digitally available, ease communication further and improve mobility, it seems reasonable to implement them in the next iteration.

C. Discussion

It has to be noted that the team leaders currently only investigated the app exploratorily, and a study in which we observe how they really use the app in their daily work will provide a clearer picture and will be the next step. Nevertheless, the results taken from this study are valuable and important in the user-centered design process. They revealed certain usability issues that need to be targeted first, before an unsupervised study can provide reliable results. On the other hand, we learned that the integrated features are perceived as valuable and that the team leaders believe that these can ease their work. The implementation of requested features will improve the mobile application and increase the chance that it will become a helpful tool in everyday work. A further analysis of the corresponding old and new interactions with the Teamleader App will lead to deeper scientific investigations.

IX. Conclusion and Future Work

In this work, a mobile application for team leaders, to support them in making ad-hoc planning decisions, was proposed. Therefore, the interaction design process started with a requirements analysis, which led to five use cases. Addressing these use cases, a system was developed that follows a classical client-server architecture. The server provides access to data stored in a graph data base that contains the domain model of the suggested solution. Further information is fetched from existing systems, such as ERP systems and a production simulation component that computes predictions about the daily production for a given staff plan. The conducted case study showed that team leaders see a benefit in the current prototype. It also showed that there is still room for improvement to make the application usable without explanation. A set of additional features was identified which could significantly improve the use of the Teamleader App in everyday work.

In future work, it is planned to improve the interaction concept of the mobile application. The usability issues identified in the case study form the entry point for this work. Further on, a set of new features that were suggested by team leaders will be included. After these improvements are realized, a discussion with the team leaders will follow to check if the improvements match their intention. Then, an evaluation is planned where team leaders use the application over a period of time during their work. It will show the benefit of the application in the real production context. In parallel, it is planned to evaluate the quality of production simulation. Historic data has been recorded that will be used to measure the correctness of values simulated by the system.

Acknowledgment

This research was funded in part by the German Federal Ministry of Education and Research (BMBF) under grant number 01IS13015 (project SmartF-IT). The responsibility for this publication lies with the authors.

References

[1] S. Knoch, M. Reiplinger, and R. Vierfuß, "Mobile Staff Planning Support for Team Leaders in an Industrial Production Scenario," in *Proceedings of the Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014), Rome, Italy*. International Academy, Research and Industry Association (IARIA), 2014, pp. 44–47.

[2] P. Lessel, M. Müller, and A. Krüger, "Towards a Novel Issue Tracking System for "Industry 4.0" Environments," in *CHI '15 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '15. New York, NY, USA: ACM, 2015.

[3] V. Pavlov, S. Knoch, and M. Deru, "CEPBoard Collaborative Electronic Performance Board and Editor for Production Environments in Industry 4.0," in *Proceedings of the International Symposium on Pervasive Displays. International Symposium on Pervasive Displays (PerDis-15), June 10-12, Saarbrcken, Germany*. ACM, 2015.

[4] H. Kagermann, W. Wahlster, and J. Helbig, Eds., *Securing the Future of German Manufacturing Industry: Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0, Final Report of the Industrie 4.0 Working Group*. Berlin: Forschungsunion im Stifterverband für die Deutsche Wirtschaft e.V., Apr. 2013. [Online]. Available: http://forschungsunion.de/pdf/industrie_4_0_final_report.pdf

[5] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & Information Systems Engineering*, vol. 6, no. 4, pp. 239–242, Aug. 2014.

[6] F. Mattern and C. Floerkemeier, "From the Internet of Computers to the Internet of Things," in *From Active Data Management to Event-Based Systems and More*, ser. Lecture Notes in Computer Science, K. Sachs, I. Petrov, and P. Guerrero, Eds. Berlin: Springer, 2010, vol. 6462, pp. 242–259.

[7] W. Wahlster, "Semantic Technologies for Mass Customization," in *Towards the Internet of Services: The THESEUS Research Program*, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Berlin: Springer, 2014, pp. 3–13.

[8] J. Preece, Y. Rogers, and H. Sharp, *Interaction Design: Beyond Human-Computer Interaction*, 4th ed. Hoboken, NJ: Wiley, 2015.

[9] J. Grudin, "Computer-supported Cooperative Work: History and Focus," *Computer*, vol. 27, no. 5, pp. 19–26, May 1994.

[10] P. Wilson, G. B. T. C. Computer, and T. A. A. C. Branch, *Computer Supported Cooperative Work: An Introduction*. Springer, 1991. [Online]. Available: http://books.google.de/books?id=-EMqvaITnE0C

[11] R. Johansen, *GroupWare: Computer Support for Business Teams*. New York, NY, USA: The Free Press, 1988.

[12] V. M. R. Penichet, I. Marin, J. A. Gallud, M. D. Lozano, and R. Tesoriero, "A Classification Method for CSCW Systems," *Electron. Notes Theor. Comput. Sci.*, vol. 168, pp. 237–247, Feb. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.entcs.2006.12.007

[13] J. Kletti, *Manufacturing Execution Systems (MES)*. Berlin; London: Springer, 2007.

[14] RWTH Aachen University IAW, "ENgAge4Pro Project," 2014, URL: http://www.iaw.rwth-aachen.de/index.php?article_id=20&projid=89&proalias=ENgAge4Pro [accessed: 2014-04-10].

[15] Fraunhofer IAO, "EPIK Project," 2014, URL: http://www.epik-projekt.de/ [accessed: 2014-04-10].

[16] ——, "KapaflexCy Project," 2014, URL: http://www.kapaflexcy.de/ [accessed: 2014-04-10].

[17] E. Hull, K. Jackson, and J. Dick, Eds., *Requirements Engineering*. London: Springer, 2011.

[18] A. Cockburn, *Writing Effective Use Cases*. Boston, MA: Addison-Wesley, 2001.

[19] R. Pressman, *Software Engineering: A Practitioner's Approach*, 6th ed. McGraw-Hill, 2005.

[20] A. Spaulding and J. S. Weber, "Usability Engineering Methods for Interactive Intelligent Systems," *AI Magazine*, vol. 30, no. 3, pp. 41–47, 2009.

[21] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-Centered Design," in *Berkshire Encyclopedia of Human-Computer Interaction, Volume 2*, W. S. Brainbridge, Ed. Great Barrington, MA: Berkshire, 2004, pp. 763–768.

[22] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-oriented Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1995.

[23] I. Fette and A. Melnikov, "The WebSocket Protocol," http://tools.ietf.org/html/rfc6455, 2011, [Accessed 27-February-2015].

[24] J. Haupert, "DOMeMan: A Framework for Representation, Management, and Utilization of Digital Object Memories," in *9th International Conference on Intelligent Environments (IE) 2013. International Conference on Intelligent Environments (IE-13), 9th, July 18-19, Athen, Greece*, J. C. Augusto, V. Bourdakis, D. Braga, S. Egerton, K. Fujinami, G. Hunter, F. Kawsar, A. Lotfi, D. Preuveneers, A. W. bin Abdul Rahman, and V. Zamudio, Eds. IEEE, 7 2013, pp. 84–91.

[25] Neo Technology, Inc., "Neo4j," http://www.neo4j.org/, [Accessed 23-June-2014].

[26] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C, Tech. Rep., 2004. [Online]. Available: http://www.w3.org/TR/owl-features/

[27] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*. Beijing: O'Reilly, 2013. [Online]. Available: http://my.safaribooksonline.com/9781449356262

[28] Siemens Product Lifecycle Management Software Inc., "Plant Simulation," 2015, URL: http://www.plm.automation.siemens.com/en_us/products/tecnomatix/manufacturing-simulation/material-flow/plant-simulation.shtml [accessed: 2015-02-24].

[29] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-Centered Design," in *Encyclopedia of Human-Computer Interaction*, W. S. Bainbridge, Ed. Berkshire Publishing Group, 2004, pp. 763–767.

[30] C. Lewis and J. Rieman, "Task-Centered User Interface Design: A Practical Introduction," Online, 1994, http://hcibib.org/tcuid, last accessed on March 1, 2015.

[31] S. Oh and B. M. Wildemuth, "Think-aloud Protocols," in *Applications of Social Research Methods to Questions in Information and Library Science*, B. M. Wildemuth, Ed. Libraries Unlimited, 2009, ch. 19, pp. 178–188.

[32] J. Brooke, "SUS: A Quick and Dirty Usability Scale," in *Usability Evaluation in Industry*, P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland, Eds. Taylor and Francis, 1996, pp. 189–194.

[33] J. Nielsen and R. Molich, "Heuristic Evaluation of User Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '90. New York, NY, USA: ACM, 1990, pp. 249–256. [Online]. Available: http://doi.acm.org/10.1145/97243.97281

[34] J. R. Lewis and J. Sauro, "The Factor Structure of the System Usability Scale," in *Proceedings of the 1st International Conference on Human Centered Design: Held as Part of HCI International 2009*, ser. HCD '09, San Diego, USA, 2009, pp. 94–103. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02806-9_12

A Comparative Study of Keyword-Based Search Features in Content-Oriented Networks

Kévin Pognart, Yosuke Tanigawa, and Hideki Tode
Dept. of Computer Science and Intelligent Systems
Osaka Prefecture University
Osaka, Japan
{pognart@com., tanigawa@, tode@}cs.osakafu-u.ac.jp

***Abstract*— The Internet shows limited performances for users' needs, especially on content sharing and video streaming. Content-Oriented Networks (CONs) are efficient approaches for such uses. They abandon the location-based routing of the Internet (IP routing) for a content identifier-based routing. In CONs, users must know the exact content identifier to request it. To give users an easier use of CONs, we quantitatively compare two keyword-based search features for CON: the existing Independent Search and Merge (ISM) and Keyword-based Breadcrumbs (KBC) we propose. While ISM uses routers to store mapping information between a content and its locations, and between a keyword and its corresponding contents, the proposed KBC simply uses routers to store information about contents went through them, in CONs based on Breadcrumbs (BC). We present in this paper the working schemes of ISM and KBC, and we compare their advantages and inconvenience, and their performances using simulation results.**

Keywords-Keywords-based Breadcrumbs; Independent Search and Merge; Content-Oriented Network; search; keyword; cache.

I. INTRODUCTION

The current Internet was made for an efficient communication between two machines by its host-to-host architecture. Nowadays, main use of the Internet is to watch video streams and to share contents, but the host-to-host architecture has limited performances. That is the reason why we present Keyword-based Breadcrumbs (KBC) [1], another network architecture with a user-friendly content searching feature exploiting its specificities. The inspiration comes from peer-to-peer (BitTorrent), which improves content sharing performances by coordinating several users by the contents they have. Content-Oriented Network (CON), which is a network architecture based on peer-to-peer features, is an alternative to the current Internet. In CON, messages are routed using content identifier instead of location identifier. In-network caches can store copies of contents while keeping the same content identifier. A content and its copies are considered identical when requesting it.

As for CON, several routing methods have been proposed to realize it [2][3]. In our work, we particularly focus on Breadcrumbs (BC) [4][5] due to its attractive features described in Section II. BC is a feasible hybrid solution that simply provides content-oriented capability over the current IP network. This BC-based CON has simple content caching, location and routing systems. In BC, we assume that users and possibly routers have a content cache. Routers have also a BC table used to route requests. When content passes through a router, a BC entry is created in its BC table to indicate the direction of the cached content. If the content goes through a node having a content cache, the content is cached. Requests are firstly sent to a server to download contents by using IP routing. When a request arrives at a router where a BC entry for the same content identifier exists, the request is redirected to follow the direction shown in the BC entry. Each next node will redirect the request according to the direction in BC entries until finding the content in a content cache. If an issue occurs during the redirection, the BC entries are invalidated and the request is forwarded again to the server by IP routing.

To perform the routing, content identifiers must be unique. This uniqueness makes the requests difficult from a user's point of view. This problem also exists in the current Internet with URLs, and it leads to the need to use web search engines. Current web search engines are not an efficient solution because they use location and they cannot use cached content information. Hence, we propose KBC [1]. We extensively designed the BC framework to complement it with a keyword-based search feature while keeping the way of working of BC and its advantages. We introduced different KBC request behaviors to retrieve answers. Also, we have implemented another keyword-based retrieval function called Independent Search and Merge (ISM) [6] for comparing their architecture and their performances.

In this paper, we present CONs. Then, we propose principle, specifics, and settings of KBC. After that, we describe also principle, specifics, and settings of ISM. We evaluate KBC and ISM performances by comparing them with some simulation scenario. After, we summarize the advantages and inconveniences of KBC and ISM, we conclude about our choice to work on KBC, and we talk about our future work.

II. RELATED WORK

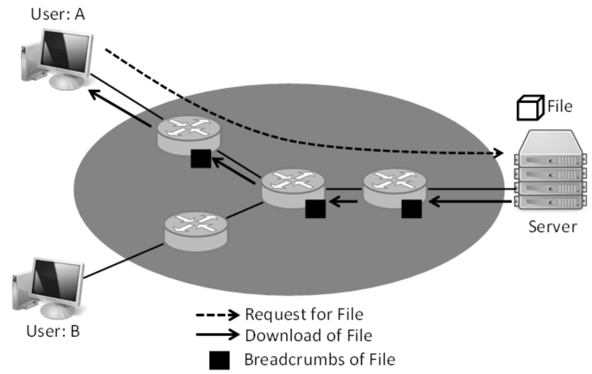
Our work is an enhancement of CONs from a user's point of view. Hence, we compared several CONs and chose the one with interesting characteristics.

A. Related CON schemes

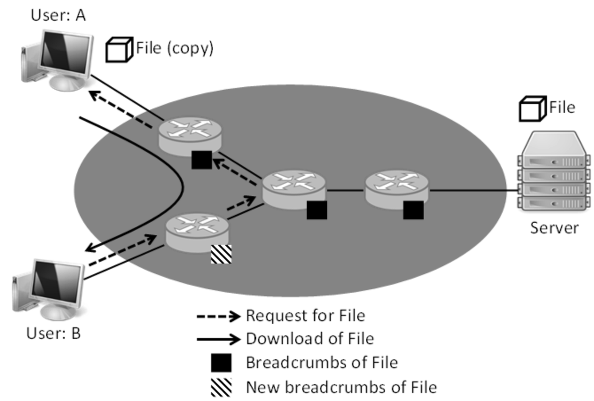
To create a CON, several schemes have been proposed. The Data oriented Network Architecture (DONA) [7], the Network of Information (NetInf) [8], the Publish-Subscribe Internet Routing Paradigm (PSIRP) [9], and the Content-Centric Networking (CCN) [10] are the main approaches. In DONA, sources publish contents into the network and their information is spread to the nodes called resolution handlers. A request goes to a resolution handler to be routed to the content. Then, the content is sent back to the requester by the reverse path or by a shortest route. NetInf can retrieve contents by name resolution and by name-based routing. Depending on the model used, the publication of a content uses a Name Resolution Service (NRS) by registering the link between the name and the locator, or it uses a routing protocol to announce the routing information. A node having a content copy can register it with NRS and by adding a new name/location binding. If an NRS is available, the requester can first resolve a content name into several available locators and find a copy from the best source. Alternatively, the requester can send a request with the content name for finding a content copy by name-based routing. Then, content found is sent back to the requester. In PSIRP, contents are published into the network but publications receive a particular Name Scope. Users can subscribe to contents. Publications and subscriptions are linked by a rendezvous system. It uses the scope identifier requested and the rendezvous identifier to form the name of the content. And by a matching procedure, the corresponding forwarding identifier is sent to the content source. Then, the content is sent to the requester. In CCN, contents are published at servers and nodes, and routing protocols are used to distribute the content location information. Requests are forwarded toward a publisher location. CCN router maintains a Pending Interest Table (PIT) for outstanding requests. PIT maintains this state for all requests and maps them to the requester network interfaces. Contents are then sent to the requester interfaces. CCN can perform on-path caching: when a content arrives at a router, this router can cache a content copy. It allows subsequent received requests for that content to be answered from that cache. While the namespace of DONA, NetInf and PSIRP are flat and names are not human-readable, the CCN namespace is hierarchical and the names can be human-readable. Flat namespace allows persistent names while the hierarchical one is IP compatible. With flat namespace, the routing is structured and the control overhead is low. With hierarchical namespace, the routing is unstructured based on flooding and the control overhead is high.

B. Breadcrumbs-based CON

We particularly focus on Breadcrumbs [4][5], which has been designed to reduce server loads and to form an autonomous CON in cooperation with cached contents. The network is a cache network where routers can cache contents and manage a table of BC entries, which are guidance information to a node holding the corresponding content. Note that in our research, actually, not core nodes but edge nodes including STBs or terminals only have content caches



(a) Download of a content from a server



(b) Download of a content using Breadcrumbs information

Figure 1. Breadcrumbs system overview

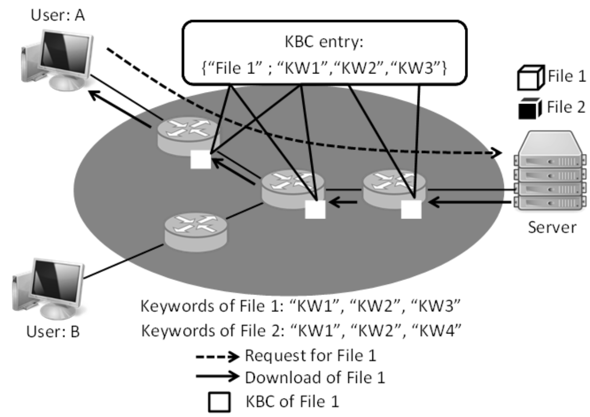
for higher feasibility, though this limitation can be removed easily. When a content passes through a router, this router creates in its BC table a BC entry corresponding to the content as shown in Figure 1 (a). A BC (BC entry) is data containing the content ID, the next node and the previous node on the content path, and the most recent time at which the content was requested and was forwarded via this router. BC is used for in-network guiding of request. Nodes information in BC is used to route requests. Time information is used to manage BCs in BC table and delete the outdated ones (since the last time update if any). When a request is created at a user node, its destination is set to a server containing the desired content in an ideal case. On its path, if the request encounters a router where a BC corresponding to the desired content exists, the router will redirect the request to the direction of the next node indicated by the BC entry, and the subsequent BC trail, series of BC entries, will guide the request until it finds the content in a cache as shown in Figure 1 (b). If a problem occurs during this redirection (content not cached at BC trail destination, lack of BC entry in the BC trail), the request is redirected to its initial server by IP routing while invalidating the whole corresponding BC entries. Hence, BC trails can be followed in both directions: one is used for finding content, and the other one is used to invalidate the BC trail. Namely, through tracing a series of BC entries, a request can follow the content downloaded previously. Some advantages are that the server loads are reduced and that there is no need to implement coordination protocol for cached contents. Also, it combines IP-routing for the first destination of request and BC trail routing when a right BC is found on path by requests. In terms of feasibility and scalability, BC is very interesting. It combines location-based routing and content name-based routing. Moreover, since location-based routing is the default routing system, BC can work in a partial deployment scenario allowing incremental deployment in the network. It has been demonstrated that this partial deployment is feasible but the performances highly depend on the deployment proportion [11]. Nevertheless, it has been shown that overlay can be used to improve these performances too.

C. Unknown contents search feature in CON

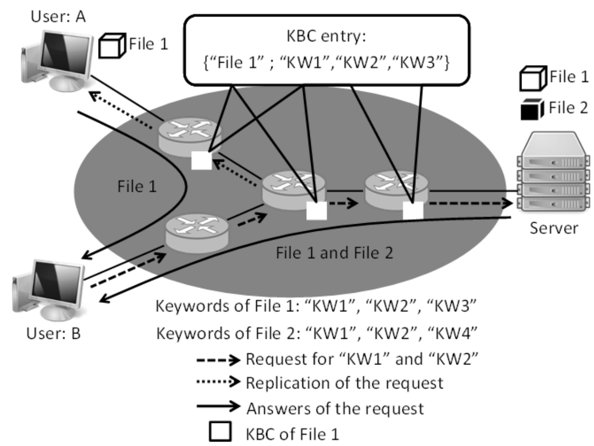
Regarding the keyword-based search feature for CON, some approaches have been proposed. It is important to add this feature in CON because the current web search engines use centralized data centers, and they cannot access caches. Hence, some advantages due to basic concepts of CON are not used. One approach to provide such a feature is to implement a system similar to typical multimedia search engines into CCN [12]. This system searches the contents similar to the content the user includes inside its request. When a search by content name is performed, the search interest is flooded over the network. Each node sharing searchable content performs a feature extraction task: a feature vector containing information about the content characteristics is extracted for each content, and an index is formed for each content type. When a request for similar contents is received by a node, it performs similarity search by comparing the request descriptors (the feature vector of the content requested) against the index to find a set of the most similar contents. When a node has similar contents, a new content is created: it is a collection of corresponding CCN links constituted by a label name for the similar object descriptors and a target name for the CCN name. An interest is sent to the requester to inform him about its availability. He requests the collection objects from each interest received. Then, data packets carrying the collections of names of similar objects are sent to the requester. This approach seems to be extendable to a keyword-based search feature but it does not seem feasible for a network such as the current Internet because of the flooding of messages all over the network.

Another approach consists in giving each keyword (or group of keywords) a numerical keyword ID as in Independent Search and Merge and Integrated Keywords Search [6]. Using these keyword IDs, the system can retrieve the answers for a keyword-based request by finding contents related to the keyword. This information is stored in router tables. We will develop Independent Search and Merge mechanisms in Section IV.

The approach we work on is Keyword-based Breadcrumbs [1]. The main idea is to use the specificities of Breadcrumbs [4], content information stored in routers on-path, to store also keywords related to content in routers on the content path. When a keyword-based request arrives at a router, all content information is checked and if some contents have all the requested keywords, the request is copied to find these contents using BC specifics. These mechanisms will be explained in the next section.



(a) Download of a content from a server



(b) System behavior to a keyword-based request

Figure 2. Keyword-based Breadcrumbs system overview

III. KEYWORD-BASED BREADCRUMBS – OUR PROPOSAL

In order to have a feasible and scalable keyword-based search feature for CON, we propose Keyword-based Breadcrumbs (KBC). Our goal is to add an intrinsic keyword-based search feature to BC system while preserving the BC advantages in terms of simplicity, scalability, feasibility and working. For this purpose, we add elements to BC system to allow two ways of working: the standard working using content name-based request and sending back of content, and a new one using keyword-based request, where KBC entries are used to find other contents in other location than server, and where answers are information about content and not the content itself. To distinguish BC system and KBC system, BC entry will be renamed to KBC entry from now when it concerns KBC system.

A. Principle of the Keyword-Based Search Feature

The basic idea is to use KBC to find closest corresponding contents. In the initial state, there are no cached contents and no guidance information. When a content is downloaded, KBCs are created on-path like in the

BC system as shown in Figure 2 (a). The difference appears for a keywords-based request. For the KBC request, the first destination remains a server. If the request reaches a node with one or more KBC entries whose keywords correspond to the requested ones, the request will be replicated as shown in Figure 2 (b). Replicated requests follow their KBC trail while the original request continues its path to the server. Then, when a right content is found, an answer containing the content ID, its list of keywords and its location is sent back to the requester. By this method, the requester can get a large number of answers with information for choosing the one he wants and if there are several identical contents, he can select the closest one. Also, IP-routing is used for downloading a content found by such a request because the answer gives the content ID and the location, and so performing another BC request for this content ID is unnecessary.

B. *Specificities of KBC*

In the proposed KBC system, we created new messages type: requests by keywords (KBC request, in opposition to BC request for a request by content ID) and answer (to KBC request because for BC request the answer is the content itself). We set rules for managing the behavior of KBC request. Also, some additions have been done to nodes to allow the use of keywords. As described previously, content has its list of keywords in addition to its ID for the creation of KBCs. Each server contains contents and a server table, which contains some of its closest other servers. This information is used to redirect KBC request for having enough answers. KBC entry contains the content ID, the content keywords, the next node and the previous node on the content path, and the most recent time the content was requested by its ID and was seen at this node. Time information is used to manage KBC in KBC table. If a KBC timer reaches the time out limit because of inactivity, it is deleted. KBC request contains the list of keywords set by the requester, a request ID for managing answers and for avoiding the previous issue, and the last node ID on its path. An answer contains the content ID and its list of keywords for allowing the user to know if this answer corresponds to the content he wants. Also, it contains the request ID for linking the answer to its request. And it contains the location of the content for allowing the requester to select the closest one he wants between several identical contents. Note that an answer must not contain the content itself. The goal is to search corresponding contents, but the user has to select the content(s) he wants from the answers list before the download. Thus, answers to KBC requests are only information about contents and not the contents themselves. We also made some optimization for reducing the number of request replications:

- Routers have a KBC table and a KBC request table containing the request IDs of recent KBC requests that went through them. This table exists to avoid an issue of KBC request replications loop. This issue happens when a triangle of routers is as follows: one KBC trail follows two edges and another one with the same keywords list follows the last edge in the same way. Figure 3 presents such a situation. The two nodes containing these two KBC entries (nodes 2 and 3) create KBC request replication whenever KBC request for the corresponding keywords list goes in. In node 2, a replication is made for the black KBC trail, and in node 3 a replication is made for the white KBC trail, which will go again in node 2 via node 1, and so on.
- In KBC request, the information of the last node ID on path is used to reduce useless replications. When a request follows even partially a KBC trail on its reverse path, each router will replicate the request to follow this KBC trail as shown in Figure 4. Then, a lot of replications are created for only a single KBC trail. Only the first replication to follow the KBC trail is enough, others flood the network. Hence, if the next node shown in KBC is equal to the previous node on the request path, the request is not replicated.
- In KBC answer, the addition of the content location assures to choose the closest content between the answers. And also, the following request for a content found by this KBC search will not perform another search in the network (BC request) because this work was already done with the KBC search.

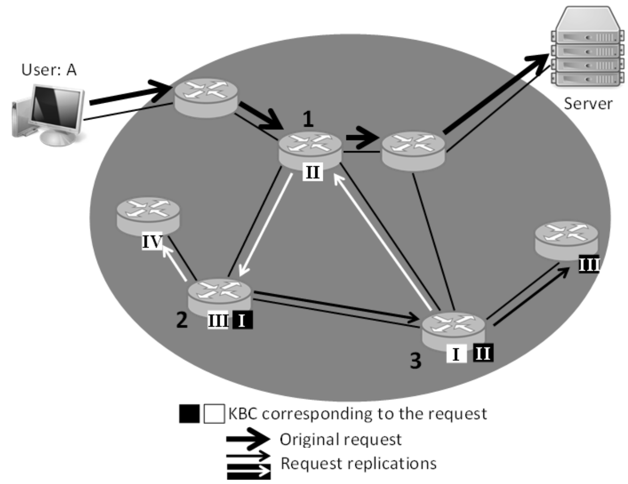


Figure 3. KBC request replications loop

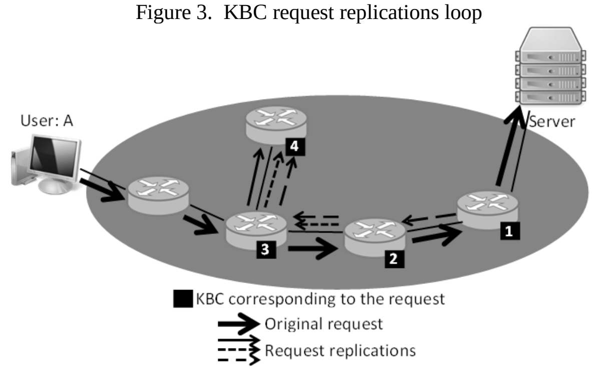


Figure 4. KBC request flooding by following a KBC trail

C. KBC request settings

An important challenge for keyword-based search feature in CON is to be efficient while not overloading the network and limiting the messages flooding. We propose here two KBC request settings to manage their behavior. As explained previously, a request needs a server destination at its creation. In “1 Server”, the KBC request is sent to one server only, but we set a threshold of minimum number of answers found in server. If this threshold is not reached, the request is redirected to another server, which was not reached yet by the request, thanks to the servers table. In “1 Server Extended”, we keep the settings from “1 Server” but we propose to add new information in contents and KBCs, the origin server location of the content. If a KBC request finds a KBC entry whose origin server is not one of the destination servers, a request replication is created to go to the new server. For “1 server extended”, we also made some optimizations:

- We add also in KBC request a list of destination servers, which is updated at each replication for a server to avoid useless replications.
- We add to server a past history request ID table to avoid several answers for the same KBC request ID. The server sends answers for a KBC request only one time for each KBC request ID.
- For a KBC request, original destination server and other destination servers found on-path are distinguished. The threshold of minimum number of answers found in servers is tried to be reached only with the original destination server.

IV. Independent Search and Merge – Compared Method

To evaluate the efficiency of KBC, we implemented another similar existing approach: Independent Search and Merge (ISM) [6]. In ISM, content ID and keywords describing the content are used to publish content. During a keyword-based search, some routers retrieve content IDs corresponding to the requested keywords. Then, at the user device, only the intersection of all content IDs retrieved are shown as the answers of the search. Then, user can perform a search using the content ID he selected.

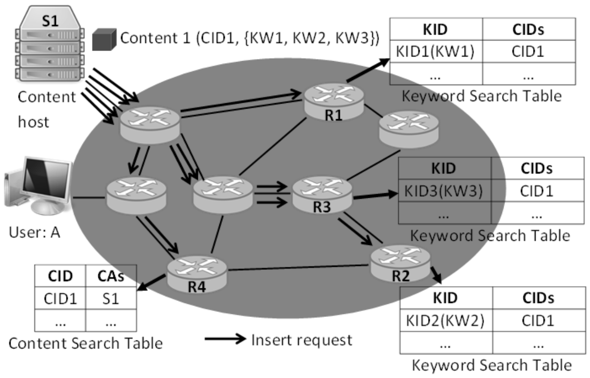
A. Principle of Independent Search and Merge Feature

The main idea is to link content ID to content locations, and to link keyword to corresponding content IDs. Hence, knowing a content ID allows the network to find easily all locations of the content. Also, by searching for some keywords, the network can retrieve easily content IDs corresponding to each keyword, and by taking into account only the intersection of all lists of answers, the user has its final list of results. One characteristic point in ISM is that each keyword is managed independently. To store all this information, routers are used. They have 2 tables: A Content Search Table, which stores for each content ID the corresponding content locations, and a Keyword Search Table, which stores for each keyword the corresponding content IDs. Also, using some algorithms, these 2 tables are not redundant between all routers. Each router is assigned to specific keywords and content IDs.

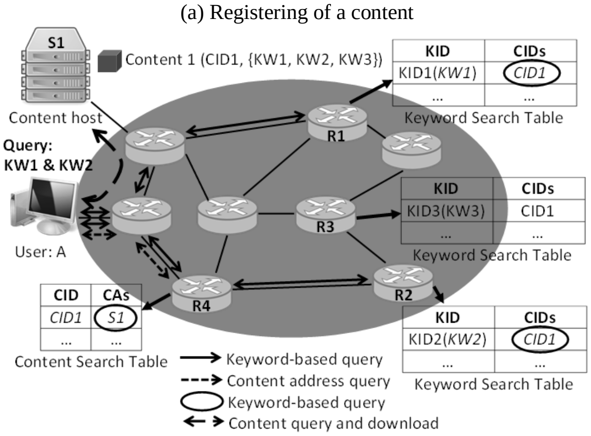
Registering a content: To register a content, we need to map the content ID and the content location, and we need to map each keyword and the content ID. For this purpose, we convert each keyword to a keyword ID, similar to content ID but belonging to a different domain, by using a pre-defined hash function. Also, another hash function is used to map any ID (content ID or keyword ID) to an IP address. The resulting IP address must correspond to an existing router IP address. If it is not the case, the hash function is applied again until generating a valid IP address. The generation of a valid IP address is assured by the use of DMap [13]. Regarding the content location, the previous hash function is applied to the content ID to generate the router IP address corresponding to this content ID. Then, an insert request containing the content ID and the content location is sent to this router. Once the router receives this insert request, the new mapping entry is added in the router Content Search Table. Regarding the keywords, each keyword is converted into a keyword ID, and for each keyword ID a router IP address is generated. For each keyword ID, an insert request containing the keyword ID and the content ID is sent to the corresponding router. Once the router receives this insert request, the mapping information is added to the router Keyword Search Table. Because content ID and keyword ID domains are different, routers can distinguish if the insert request concerns the Content Search Table or the Keyword Search Table. This scheme is explained in Figure 5 (a), where Content 1 is uploaded in the server S1. Once the upload is done, the content needs to be registered. Hence, S1 creates an insert request to link Content 1 ID CID1 and its location S1, and 3 insert requests to link the keywords KW1, KW2 and KW3 to CID1. For the first insert request, the hash function is used on CID1, and the result is the IP address of the router R4. Hence, this insert request is sent to R4. For the keywords, they are first converted to keyword IDs KID1, KID2, and KID3 by another hash function, then the previous hash function is used. KID1 leads to R1, KID2 leads to R2, and KID3 leads to R3.

1) Requesting a content by its content ID: To request a content using its content ID, we just need to apply the hash function to the content ID to obtain the router IP address where the information about the content locations is stored. A request containing the content ID and the requester IP address is sent to the corresponding router. Then the router sends back the content locations found in its Content Search Table. Finally, the requester needs to send a similar request to one of the content locations answered to download the content.

2) Keyword-based request: The user makes a request by using several keywords. At the user node, each keyword is



(a) Registering of a content



(b) System behavior to a keyword-based request followed by a content download request

Figure 5. Independent Search and Merge system overview

converted to a keyword ID using the same pre-defined hash function as the one used for registering contents. Then, for each keyword ID, the other hash function is applied to have the IP address of routers having information about these keywords. Hence, for each keyword ID, a request containing the keyword ID and the requester IP address is sent to the corresponding router IP address. At the router, the list of content IDs corresponding to the keyword ID is retrieved from the router Keyword Search Table and sent to the requester. Once the requester received all answers, only the content IDs shared by all lists are kept and shown to the requester. Finally, the requester can select one of the content IDs to perform a request using the content ID. This behavior is explained in Figure 5 (b). User A makes a keyword-based request with the keyword KW1 and KW2. They are converted into the keyword IDs KID1 and KID2. By using the hash function, KID1 leads to the router R1, and KID2 leads to the router R2. The content ID information is sent back to user A: CID1 by KID1, and CID1 by KID2. Only the content IDs in common are kept: CID1. Then user A requests the content CID1. By using the hash function, it leads to the router R4. CID1 location (S1) is sent back to user A. Finally, user A sends a request for CID1 at server S1, and S1 replies the content CID1.

V. Evaluation

To evaluate KBC, we used different simulation scenarios for KBC and ISM. We use the simulation results to compare them.

A. Simulation Scenario

1) General settings:

- Network Topology: To evaluate the proposed schemes, we use a flat router-network based on the Waxman model on a lattice points of 1000x1000, α=0.1 and β=0.05 [14]. There are 1000 routers, 5000 users and 50 servers. Each router is linked to five users and the server locations are chosen according to uniformly random distribution. Regarding caches, only edge nodes including STBs or terminals have content caches for higher feasibility, though this limitation can be removed easily. Each cache can have a maximum of two contents.
- Keywords: we set three types of keywords (KW1, KW2 and KW3), which are hierarchically linked. All contents and requests contain one of each previous type of keywords (1 KW1, 1 KW2 and 1 KW3). In KBC system, a KBC request is initially routed toward a server. Keyword types are hierarchical for practicability of the initial routing. KW1 represents the main characteristic of the content (video, audio, etc.). Only keywords belonging to a single KW1 are used. KW2 represents a sub-domain of KW1 (if KW1 is “Video”, KW2 can be “Action”, “News”, “Sports”, etc.). There are 25 different keywords for KW2. KW3 is a more specific keyword describing more precisely the content. For each KW2, there are four keywords possible for KW3. In total, 100 keywords combinations are possible.
- Contents: Servers contain in total 10,000 contents, which are all unique by their content ID and which are all defined by three random keywords (one of each keyword type). Hence, each keyword combination corresponds to around 100 contents. Also, servers have the same contents during all the simulation time and for each simulation.
- Requests: The two types of user request (by content ID and by keywords) are generated at an independent, identical and exponentially-distributed random interval. In a first time, 50,000 requests by content ID are made for initializing the network and for spreading contents information. Then, we study the systems for 55,000 content ID-based requests and a variable number of keyword-based requests depending on the wanted ratio between these two request types.
- Answers: When answers for keyword-based requests are received, one of them is selected to download the content.

We have four requests patterns to switch between content ID-based (CID) requests and keyword-based (KW) requests. For 1 KW request, 2 CID requests are performed (2 CID), 4 CID requests are performed (4 CID), 10 CID requests are performed (10 CID) or 15 CID requests are performed (15 CID). Also, we set four thresholds for the minimum number of answers found in servers: 5, 10, 15 and 20. For each threshold value, we average the results for each different requests pattern to focus on the threshold values.

2) Keyword-based Breadcrumbs: To evaluate KBC, we use a modified version of Breadcrumbs+ (BC+) simulator [5] for implementing KBC. Hence, BC+ with adaptive invalidation is used instead of BC. It is an improvement of BC to avoid the issue in which some requests cannot reach the intended content in a particular situation. The differences with BC are that a BC+ entry has a list of the previous nodes on the content path instead of the previous one only, and if at the end of a BC trail, content is replaced or cannot be cached, an invalidation message is sent to all the nodes in this previous nodes list.

- Servers: Each server has address information of its three nearest server neighbors to redirect the requests in the situation where the threshold number of answers from servers is not reached. Servers have also a list of request IDs of requests, which went to them. We did not set a size for this set. However, it can be easily done by setting a time out to entries.
- KBC table: It does not have limitation about its size but information about its size is collected during simulations.

3) Independent Search and Merge: In ISM, all information is in router tables. All content IDs are equally distributed between all routers, and all keyword IDs are equally distributed between all routers. The function used to link IDs to router addresses is pre-defined. Also, at the beginning of the simulation, the information of contents in servers are known by the corresponding tables (content ID-content location, and keyword ID-content ID).

B. Performances

1) Keyword-based Breadcrumbs: Figure 6 presents the efficiency of KBC system for retrieving right contents. The setting of 1 Server has limited performances because requests are restrained to a close area of the first destination server. 1 Server Extended shows high efficiency for finding different but right contents.

Figure 7 shows the number of KBC request replications. With 1 Server, requests are replicated only few times in mean, which means that it does not degrade the network performances. With 1 Server Extended, replications are numerous and can interfere with a good network working.

The repartition of answers between caches and servers shown in Figure 8 is also interesting because it indicates how many KBC trails are successfully followed. Once again in 1 Server, the results are low. On the other hand, 1 Server Extended can find a lot of corresponding KBC trails even if the threshold in server is low. In a small network area, KBC requests can easily find a KBC about a content from outside of this area. Hence with 1 Server Extended, KBC requests can go all over the network. It is confirmed by the equality between the number of different contents found (Unique Contents) and the number of contents found in servers.

Figure 9 shows how many KBC requests do not gather enough answers in servers. It indicates which thresholds are fitting or not, knowing the number of contents and their keywords. As expected, the lower the threshold is, the lower the failing rate is. But if the threshold is too low, user does not receive enough answers. About this trade-off, choosing a medium threshold (10 or 15) seems a good compromise.

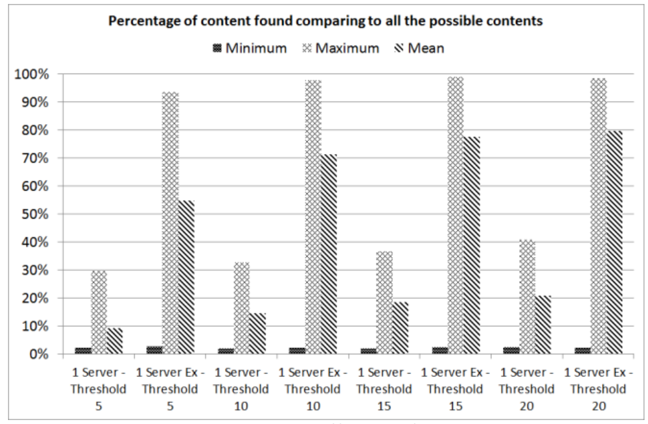


Figure 6. Content retrieval efficiency for KBC requests

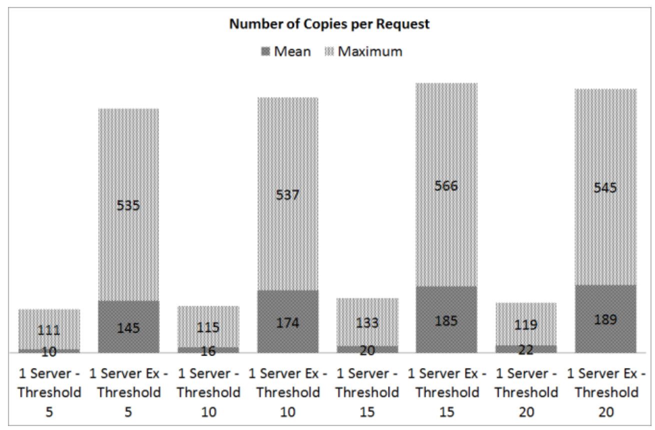


Figure 7. KBC request replications

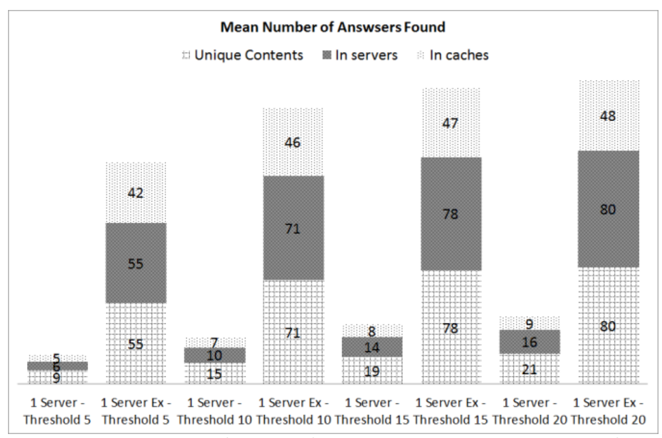


Figure 8. Repartition of answers from server or cache, and number of unique contents found (without taking into account identical contents)

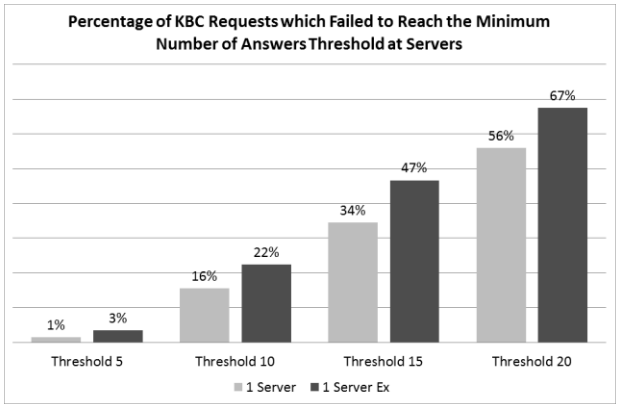


Figure 9. KBC reaching threshold failing rate

TABLE I. KBC TABLE SIZE

Mean size of KBC table	Unbiased variance of KBC table size	Standard deviation of KBC table size
71	712	28

TABLE II. NUMBER OF ENTRIES IN ISM TABLES

Mean number of entries in Content Search Table	Mean number of entries in Keyword Search Table
10	0.125

TABLE III. NUMBER OF CONTENT LOCATIONS FOUND IN ISM

Minimum number of locations found	Maximum number of locations found	Mean number of locations found	Standard deviation of number of locations found	Percentage of cases where only the server location exists
1	1694	7	49	15%

Regarding the KBC tables, no limit was set because the needed size is important to know. Table I presents the mean size of KBC table with its unbiased variance and its standard deviation. Viewing these results, we can propose to have a KBC table of 100 entries, which is 1/100 of all contents in our simulated network.

2) Independent Search and Merge: Router tables being the center of ISM mechanism, we take a look at them. Content and keyword information are distributed between all routers tables. Table II shows primary information about ISM tables. The Keyword Search Table (KST) is conditioned only by the number of keywords and the number of content using these keywords. Here, we have 125 keywords, which correspond to 125 KST entries distributed in the 1000 routers KSTs. Hence, if there are too many keywords, this table becomes too huge and unfeasable. This is a limitation of ISM. About the Content Search Table (CST), its size depends on the number of contents and the number of their locations. Here, we have 10000 contents, which correspond to 10000 CST entries distributed in the 1000 routers CSTs. Table III shows the mean number of locations for a single content and its standard deviation. At least, each content can be found at a server. For all contents, there is in mean 7 user cache locations to find it, and in only 15% of keyword-based requests, only the server location was registered.

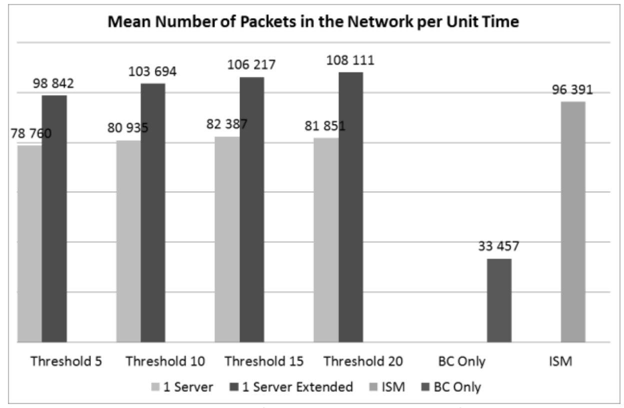


Figure 10. Mean number of packets per unit time for KBC, BC and ISM

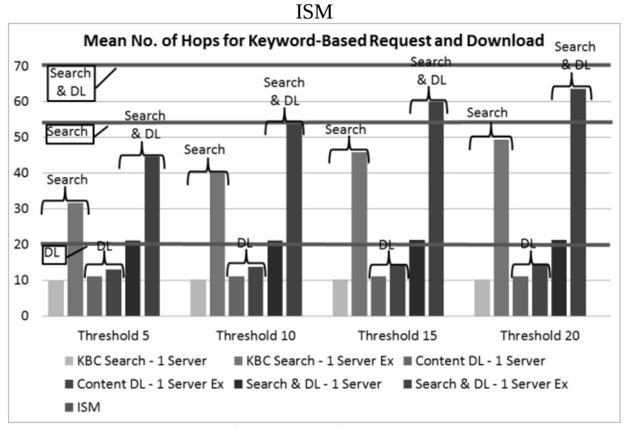


Figure 11. Mean no. of hops to perform a keyword-based request

Another aspect is the efficiency for retrieving results. ISM presents a 100% of efficiency thanks to its mechanism. However, it is counterbalanced by the necessity for routers tables to have information of all keywords and all contents.

3) Comparative results: To compare KBC and ISM, we focused on several aspects.

The mean number of packets in the network per unit time indicates which one implies the more messages creation and management in mean. As shown in Figure 10, KBC with the setting "1 Server" induces the least flooding where the setting "1 Server Extended" induces the higher flooding. ISM is between them, however, depending on the threshold, KBC "1 Server Extended" induces between 2% and 12% more packets than ISM, which is still acceptable.

Figure 11 shows the search time and the download time for a keyword-based request by using hop count. The lines correspond to ISM results. For KBC "1 Server Extended", the number of necessary hops to perform a keyword-based search depends on the minimum number of answers from server threshold. The higher the threshold is, the higher the number of hops is. Even with a high threshold, performances of ISM are worse than KBC. It can be explained by the routers locations for gathering the information. In KBC, the effective network area is small. It is close to the user and to the original destination server of the request. In ISM, the whole network is used to store the information. Also, regarding the download time in ISM, a request for the content location must be made while in KBC this piece of information is included in the keyword-based request answers.

ISM emphasis is on efficiency. Hence, all contents and all locations can be found by a keyword-based request. However, it requires router tables capacity to be high enough to store information for all existing keywords and for all contents, which can be unfeasible. Also, if a router has an issue and becomes out of order, all information stored are inaccessible and/or lost because in ISM, network-wide coordination of routers is necessary. On the other hand, KBC focuses on scalability and capability of adaptation. We want to have enough answers, but only a small part of the whole network is used. Also, the KBC information is managed automatically with time-out in tables and with invalidation messages in case of no more valid KBC trail.

VI. Conclusion

We presented in this paper a comparative study of keyword-based search features for Content-Oriented Network. Such features are important from a user point of view to make the network accessible. We based our study on the proposed Keyword-based Breadcrumbs, our keyword-based search feature based on Breadcrumbs, and on the existing Independent Search and Merge, another similar feature performing differently. They are scalable and interesting. KBC is scalable not only in CON but also in partially deployed Breadcrumbs because keyword-based search is close in its working to content name-based one, and thanks to Breadcrumbs characteristics. These features focus on different points. ISM is focused on the content retrieval efficiency. However, be careful about the size of tables used to store content information, because it depends on the number of routers, contents, and keywords. Also, a failure in the system results in a loss of data and so in efficiency. KBC is focused on the capacity of adaptation and on the user's neighborhood. Content information is stored on its path, and a keyword-based request will go to the closest server and find close content information. There is a trade-off between the network flooding and the search efficiency. It seems that KBC with the setting "1 Server Extended" is a good compromise with a high efficiency and a capacity of adaptation.

In our future work, we want to work more on KBC by changing our network for having non unique contents. Also, we will take into account the content popularity, and we want to implement an indicator of users' satisfaction (if a content is downloaded thanks to a keyword-based search, it means that for the keyword list used, the user is satisfied of this content). Also, we continue to see other possible ways to perform keyword-based search in CON.

Acknowledgment

This research was supported in part by National Institute of Information and Communication Technology (NICT), Japan.

References

[1] K. Pognart, Y. Tanigawa, and H. Tode, "Keyword-Based Breadcrumbs: A Scalable Keyword-Based Search Feature in Breadcrumbs-Based Content-Oriented Network," in Proc. AFIN 2014, Nov. 2014, pp. 20-25.

[2] J. Choi, J. Han, E. Cho, K. Kwon, and Y. Choi, "A Survey on Content-Oriented Networkinf for Efficient Content Delivery," IEEE Communications Magazine, vol. 49, no. 3, Mar. 2011, pp. 121-127.

[3] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A Survey of Information-Centric Networking," IEEE Communications Magazine, vol. 50, no. 7, Jul. 2012, pp. 26-36.

[4] E. Rosensweig and J. Kurose, "Breadcrumbs: efficient, best-effort content location in cache networks," in Proc. IEEE INFOCOM 2009, Apr. 2009, pp. 2631-2635.

[5] M. Kakida, Y. Tanigawa, and H. Tode, "Breadcrumbs+: Some Extensions of Breadcrumbs for In-network Guidance in Content Delivery Networks," in Proc. SAINT 2011, Jul. 2011, pp. 376–381.

[6] Y. Mao, B. Sheng, and M. Chuah, "Scalable Keyword-Based Data Retrievals in Future Content-Centric Networks," in Proc. 2012 Eighth International Conference on Mobile Ad-hoc and Sensor Networks (MSN), Dec. 2012, pp. 116-123.

[7] T. Koponen, et al., "A Data-Oriented (and Beyond) Network Architecture," in Proc. SIGCOMM '07, Aug. 2007, pp. 181-192.

[8] B. Ahlgren, et al., "Second NetInf Architecture Description," Deliverable D-6.2 in The Network of the Future - FP7-ICT-2007-1-216041, Apr. 2010.

[9] M. Ain, et al., "Architecture Definition, Component Descriptions, and Requirements," Deliverable D2.3 in Publish-Subscribe Internet Routing Paradigm - FP7-INFSO-IST-216173, Feb. 2009.

[10] V. Jacobson, et al., "Networking named content," in Proc. ACM CoNEXT 2009, Dec. 2009, pp. 1-12.

[11] T. Tsutsui, H. Urabayashi, M. Yamamoto, E . Rosenweig, and J. Kurose, "Performance Evaluation of Partial Deployment of Breadcrumbs in Content Oriented Networks," in Proc. IEEE ICC FutureNet 2012, Jun. 2012, pp. 5828-5832.

[12] P. Daras, T. Semertzidis, L. Makris, and M. Strintzis, "Similarity Content Search in Content Centric Networks," in Proc. ACM MM '10, Oct. 2010, pp. 775-778.

[13] T. Vu, et al., "DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet," in Proc. IEEE ICDCS 2012, Jun. 2012, pp. 698-707.

[14] B. M. Waxman, "Routing of Multipoint Connections," IEEE J. Sel. Areas Comms (Special Issue on Broadband Packet Communication), vol. 6, no. 9, Dec. 1998, pp. 1617-1622.