International Journal on Advances in Internet Technology

2016 vol. 9 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 9, Number 1 & 2, 2016

Ajay Chakravarthy, University of Southampton IT Innovation Centre, UK
Chin-Chen Chang, Feng Chia University, Taiwan
Ruay-Shiung Chang, National Dong Hwa University, Taiwan
Tzung-Shi Chen, National University of Tainan, Taiwan
Xi Chen, University of Washington, USA
IlKwon Cho, National Information Society Agency, South Korea
Andrzej Chydzinski, Silesian University of Technology, Poland
Noël Crespi, Telecom SudParis, France
Antonio Cuadra-Sanchez, Indra, Spain
Javier Cubo, University of Malaga, Spain
Sagarmay Deb, Central Queensland University, Australia
Javier Del Ser, Tecnalia Research & Innovation, Spain
Philipe Devienne, LIFL - Université Lille 1 - CNRS, France
Kamil Dimililer, Near East Universiy, Cyprus
Martin Dobler, Vorarlberg University of Applied Sciences, Austria
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Matthias Ehmann, Universität Bayreuth, Germany
Tarek El-Bawab, Jackson State University, USA
Nashwa Mamdouh El-Bendary, Arab Academy for Science, Technology, and Maritime Transport, Egypt
Mohamed Dafir El Kettani, ENSIAS - Université Mohammed V-Souissi, Morocco
Marc Fabri, Leeds Metropolitan University, UK
Armando Ferro, University of the Basque Country (UPV/EHU), Spain
Anders Fongen, Norwegian Defence Research Establishment, Norway
Giancarlo Fortino, University of Calabria, Italy
Kary Främling, Aalto University, Finland
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Ivan Ganchev, University of Limerick, Ireland
Shang Gao, Zhongnan University of Economics and Law, China
Kamini Garg, University of Applied Sciences Southern Switzerland, Lugano, Switzerland
Rosario Giuseppe Garroppo, Dipartimento Ingegneria dell'informazione - Università di Pisa, Italy
Thierry Gayraud, LAAS-CNRS / Université de Toulouse / Université Paul Sabatier, France
Christos K. Georgiadis, University of Macedonia, Greece
Katja Gilly, Universidad Miguel Hernandez, Spain
Feliz Gouveia, Universidade Fernando Pessoa - Porto, Portugal
Kannan Govindan, Crash Avoidance Metrics Partnership (CAMP), USA
Bill Grosky, University of Michigan-Dearborn, USA
Jason Gu, Singapore University of Technology and Design, Singapore
Christophe Guéret, Vrije Universiteit Amsterdam, Nederlands
Frederic Guidec, IRISA-UBS, Université de Bretagne-Sud, France
Bin Guo, Northwestern Polytechnical University, China
Gerhard Hancke, Royal Holloway / University of London, UK
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Quang Hieu Vu, EBTIC, Khalifa University, Arab Emirates
Hiroaki Higaki, Tokyo Denki University, Japan
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST), Korea
Anna Hristoskova, Ghent University - IBBT, Belgium
Ching-Hsien (Robert) Hsu, Chung Hua University, Taiwan
Chi Hung, Tsinghua University, China
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Raj Jain, Washington University in St. Louis , USA
Edward Jaser, Princess Sumaya University for Technology - Amman, Jordan
Terje Jensen, Telenor Group Industrial Development / Norwegian University of Science and Technology, Norway

Yasushi Kambayashi, Nippon Institute of Technology, Japan
Georgios Kambourakis, University of the Aegean, Greece
Atsushi Kanai, Hosei University, Japan
Henrik Karstoft , Aarhus University, Denmark
Dimitrios Katsaros, University of Thessaly, Greece
Ayad ali Keshlaf, Newcastle University, UK
Reinhard Klemm, Avaya Labs Research, USA
Samad Kolahi, Unitec Institute Of Technology, New Zealand
Dmitry Korzun, Petrozavodsk State University, Russia / Aalto University, Finland
Slawomir Kuklinski, Warsaw University of Technology, Poland
Andrew Kusiak, The University of Iowa, USA
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Frédéric Le Mouël, University of Lyon, INSA Lyon / INRIA, France
Juong-Sik Lee, Nokia Research Center, USA
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Clement Leung, Hong Kong Baptist University, Hong Kong
Longzhuang Li, Texas A&M University-Corpus Christi, USA
Yaohang Li, Old Dominion University, USA
Jong Chern Lim, University College Dublin, Ireland
Lu Liu, University of Derby, UK
Damon Shing-Min Liu, National Chung Cheng University, Taiwan
Michael D. Logothetis, University of Patras, Greece
Malamati Louta, University of Western Macedonia, Greece
Maode Ma, Nanyang Technological University, Singapore
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Olaf Maennel, Loughborough University, UK
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Yong Man, KAIST (Korea advanced Institute of Science and Technology), South Korea
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Chengying Mao, Jiangxi University of Finance and Economics, China
Brandeis H. Marshall, Purdue University, USA
Sergio Martín Gutiérrez, UNED-Spanish University for Distance Education, Spain
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Shawn McKee, University of Michigan, USA
Stephanie Meerkamm, Siemens AG in Erlangen, Germany
Kalogiannakis Michail, University of Crete, Greece
Peter Mikulecky, University of Hradec Kralove, Czech Republic
Moeiz Miraoui, Université du Québec/École de Technologie Supérieure - Montréal, Canada
Shahab Mokarizadeh, Royal Institute of Technology (KTH) - Stockholm, Sweden
Mario Montagud Climent, Polytechnic University of Valencia (UPV), Spain
Stefano Montanelli, Università degli Studi di Milano, Italy
Julius Müller, TU- Berlin, Germany
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain
Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Alex Ng, University of Ballarat, Australia
Christopher Nguyen, Intel Corp, USA
Petros Nicopolitidis, Aristotle University of Thessaloniki, Greece
Carlo Nocentini, Università degli Studi di Firenze, Italy
Federica Paganelli, CNIT - Unit of Research at the University of Florence, Italy
Carlos E. Palau, Universidad Politecnica de Valencia, Spain
Matteo Palmonari, University of Milan-Bicocca, Italy
Ignazio Passero, University of Salerno, Italy
Serena Pastore, INAF - Astronomical Observatory of Padova, Italy

Fredrik Paulsson, Umeå University, Sweden
Rubem Pereira, Liverpool John Moores University, UK
Yulia Ponomarchuk, Far Eastern State Transport University, Russia
Jari Porras, Lappeenranta University of Technology, Finland
Neeli R. Prasad, Aalborg University, Denmark
Drogkaris Prokopios, University of the Aegean, Greece
Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Lucia Rapanotti, The Open University, UK
Gianluca Reali, Università degli Studi di Perugia, Italy
Jelena Revzina, Transport and Telecommunication Institute, Latvia
Karim Mohammed Rezaul, Glyndwr University, UK
Leon Reznik, Rochester Institute of Technology, USA
Simon Pietro Romano, University of Napoli Federico II, Italy
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
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
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
Orazio Tomarchio, Universita' di Catania, Italy
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
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
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

International Journal on Advances in Internet Technology
Volume 9, Numbers 1 & 2, 2016

CONTENTS

An Experimental Evaluation of Performance Problems in HTTP Server Infrastructures Using Online Clients

Ricardo Filipe, Serhiy Boychenko, and Filipe Araujo

CISUC, Dept. of Informatics Engineering
University of Coimbra
Coimbra, Portugal
rafilipe@dei.uc.pt, serhiy@dei.uc.pt, filipius@uc.pt

***Abstract*—Ensuring short response times is a major concern for all web site administrators. To keep these times under control, they usually resort to monitoring tools that collect a large spectrum of system metrics, such as CPU and memory occupation, network traffic, number of processes, etc. Despite providing a reasonably accurate picture of the server, the times that really matter are those experienced by the user. However, not surprisingly, system administrators will usually not have access to these end-to-end figures, due to their lack of control over web browsers. To overcome this problem, we follow the opposite approach of monitoring a site based on times collected from browsers. We use two browser-side metrics for this: i) the time it takes for the first byte of the response to reach the user (request time) and ii) the time it takes for the entire response to arrive (response time). We conjecture that an appropriate choice of the resources to control, more precisely, one or two web pages, suffices to detect CPU, network and disk input/output bottlenecks. In support of this conjecture, we run periodical evaluations of request and response times on some very popular web sites to detect bottlenecks. In this paper, we present a new experiment using pairs of synchronized clients, to extend the results we achieved with single-client requests in our previous work. Results suggest that collecting timing data from the browsers can indeed contribute to detect server performance problems and raise interesting questions regarding unfair delays that seem to exist in some specific requests.**

***Keywords*–Web monitoring; Client-side monitoring; Bottleneck.**

I. INTRODUCTION

In the operation of a Hypertext Transfer Protocol (HTTP) server, bottlenecks may emerge at different points of the system often with negative consequences for the quality of the interaction with users. To control this problem, system administrators must keep a watchful eye on a large range of system parameters, like CPU, disk and memory occupation, network interface utilization, among an endless number of other metrics, some of them specifically related to HTTP, such as response times or queue sizes. Despite being very powerful, these mechanisms cannot provide a completely accurate picture of the HTTP protocol performance. Indeed, the network latency and transfer times can only be seen from the client, not to mention that some server metrics might not translate easily to the quality of the interaction with users. Moreover, increasing the number of metrics involved in monitoring adds complexity to the system and makes monitoring more intrusive. In Section II, we overview different techniques to monitor servers and to detect different sorts of bottlenecks.

We hypothesize that a simpler mechanism, based on client-side monitoring, can fulfill the task of detecting and identifying an HTTP server bottleneck from a list of three: CPU, network, or disk input/output (simply I/O hereafter). The arguments in favor of this idea are quite powerful: client-side monitoring provides the most relevant performance numbers, while, at the same time, requiring no modifications to the server, which, additionally, can run on any technology. This approach can provide a very effective option to complement available monitoring tools.

To achieve this goal, we require two metrics taken from the web browser: i) the time it takes from requesting an object to receiving the first byte (request time), and ii) the time it takes from the first byte of the response, to the last byte of data (response time). We need to collect time series of these metrics for, at least, one or two carefully chosen URLs. These URLs should be selected according to the resources they use, either I/O or CPU. As we describe in Section III, the main idea is that each kind of bottleneck exposes itself with a different signature in the request and response time series.

To try our conjecture, and create such time series, in Section IV, we resorted to experiments on real web sites, by automatically requesting one or two URLs with a browser every minute, and collecting the correspondent request and response times. With these experiments, we managed to discover a case of network bottleneck and another one of I/O bottleneck. We believe that this simple mechanism can improve the web browsing experience, by providing web site developers with qualitative results that add to the purely quantitative metrics they already own.

We now extend these results, which we presented before in [1], with an additional experiment. In Section V, we fetch pages from the same server using two synchronized clients. This enables separation between client-side network and server-side problems. However, the main goal of this experiment was to verify whether observations from one of the clients takes us into a set of conclusions that fits the observations of the second one. Furthermore, to avoid any bias, in Section VI, we introduce a simple algorithm that evaluates request and response times from both clients, before outputting the cause of the problem.

Surprisingly, we noted that, occasionally, the two clients disagree about the quality of the interaction with the server. One of them suffers from an isolated server-side problem, which does not occur again, while the other client does not

suffer from any problem at all. This suggests that some requests get a very unfair treatment along their way. Even a network and server that seem to be lightly loaded can exhibit this sort of delay at times. Determining exactly where and how frequently does this happen is, we believe, an interesting practical concern.

To summarize, this paper makes the following major contributions:

- it proposes a mechanism to detect bottlenecks on HTTP server infrastructures, based on taking periodic client-side metrics;
- it shows evidence of particularly long delays in specific isolated requests.

The rest of the paper is organized as follows. Section II presents the related work in this field and provides a comparison of different methods. Section III describes our conjecture of client-side detection and identification of HTTP server bottlenecks. In Section IV, we show monitoring results from popular web sites, thus exposing different types of bottlenecks. In Section V, we extend our previous work, now using two clients in different networks. In Section VI, we present an automated mechanism to detect bottlenecks. Finally, in Section VII we discuss the results and conclude the paper.

II. Related Work

In the literature, we can find a large body of work focused on timely scaling resources up or down, usually in N-tier HTTP server systems, [2–8]. We divide these efforts into three main categories: (i) analytic models that collect multiple metrics to ensure detection or prediction of bottlenecks; (ii) rule-based approaches, which change resources depending on utilization thresholds, like network or CPU; (iii) system configuration analysis, to ensure correct functionality against bottlenecks and peak period operations.

First, regarding analytic models, authors usually resort to queues and respective theories to represent N-tier systems [9][10]. Malkowski *et al.* [11] try to satisfy service level objectives (SLOs), by keeping low service response times. They collect a large number of system metrics, like CPU and memory utilization, cache, pool sizes and so on, to correlate these metrics with system performance. This should expose the metrics responsible for bottlenecks. However, the analytic model uses more than two hundred application and system level metrics. In [12], Malkowski *et al.* studied bottlenecks in N-tier systems even further, to expose the phenomenon of multi-bottlenecks, which are not due to a single resource that reaches saturation. Furthermore, they managed to show that lightly loaded resources may be responsible for such multi-bottlenecks. As in their previous work, the framework resorts to system metrics that require full access to the infrastructure. The number of system metrics to collect is not clear. Wang *et al.* continued this line of reasoning in [8], to detect transient bottlenecks with durations as low as 50 milliseconds. The transient anomalies are detected recurring to depth analysis of metrics in each component of the system. Although functional, this approach is so fine-grained that it is closely tied to a specific hardware and software architecture.

In [3], authors try to discover bottlenecks in data flow programs running in the cloud. In [7], Bodík *et al.* try to predict bottlenecks to provide automatic elasticity. The work in [6] presents a dynamic allocation of Virtual Machines (VMs) based on Service Level Agreement (SLA) restrictions. The framework consists of a continuous "loop" that monitors the cloud system, to detect and predict component saturation. The paper does not address questions related to resource consumption of the monitoring approach or scalability to large cloud providers. Unlike other approaches that try to detect bottlenecks, [13] uses heuristic models to achieve optimal resource management. Authors use a database rule set that, for a given workload, returns the optimal configuration of the system. The work in [14] presents a technique to analyze workloads using k-means clustering. This approach also uses a queuing model to predict the server capacity for a given workload for each tier of the system.

TABLE I. Bottleneck detection in related work.

Article	CPU/Threads/VM	I/O	Network
[3]	X	X	
[4]	X		
[11]	X	X	
[5]	X		
[8]	X	X	Internal
[12]	X	X	Internal
[17]	X		X
[15]			External

In [15], authors propose a collaborative approach. They use a web browser plug-in on each client, to monitor all Internet activity, gather several network metrics, and send the information to a central point, for processing. The focus of the plug-in is the main web (HTML) page. The impact of this approach on network bandwidth and client data security is unclear, as authors only handle external network connectivity issues.

Other researchers have focused on rule-based schemes to control resource utilization. Iqbal *et al.* [4][16] propose an algorithm that processes proxy logs and, at a second layer, all CPU metrics of web servers. The goal is to increase or decrease the number of instances of the saturated component. Reference [17] also scales up or down servers based on CPU and network metrics of the server components. If a component resource saturation is observed, then, the user will be migrated to a new virtual machine through IP dynamic configuration. This approach uses simpler criteria to scale up or down compared to bottleneck-based approaches, because it uses static performance-based rules.

Table I illustrates the kind of resource problem detected by the mentioned papers. The second column concerns the need to increase CPU resources or VM instances. The third column is associated to I/O, normally an access to a database. The network column represents delays inside the server network or in the connection to the client. It is relevant to mention that several articles [3][12][18] only consider CPU (or instantiated VM) and I/O bottlenecks, thus not considering internal (between the different components) or external (client-server) bandwidth.

Some techniques scan the system looking for misconfigurations that may cause inconsistencies or performance issues. Attariyan *et al.* [20] elaborated a tool that scans the system in real time, to discover root cause errors in the configuration. In [21], authors use previous correct configurations to eliminate unwanted or mistaken operator configuration.

It is also worth mentioning client-side tools like HTTPerf [22] or JMeter [23], which serve to test HTTP servers, frequently under stress, by running a large number of simultaneous invocations of a service. However, these tools work better for benchmarking a site before it goes online. Nevertheless, in [19], we demonstrated that it is possible to detect bottlenecks with limited access to the server using JMeter. However, the bursts of requests of JMeter could hardly work on the real internet, and could potentially be considered as a denial-of-service attack.

Our current work is different from the previously mentioned literature in at least two aspects: we are not tied to any specific architecture and we try to evaluate the bottlenecks from the client's perspective. This point of view provides a better insight on the quality of the response, offering a much more accurate picture regarding the quality of the service. While our method could replace some server-side mechanisms, we believe that it serves better as a complementary mechanism.

III. A Conjecture on Client-Side Monitoring of HTTP Servers

This section presents our conjecture concerning network, CPU and I/O bottleneck identification. The first subsection shows how to identify network bottlenecks and the second subsection shows how to distinguish CPU from I/O bottlenecks.

A. Identification of Network Bottlenecks

We now evaluate the possibility of detecting bottlenecks, based on the download times of web pages, as seen by a client. We conjecture that we can, not only, detect the presence of a bottleneck, something that would be relatively simple to do, but actually determine the kind of resource causing the bottleneck, CPU, I/O or network. CPU limitations may be due to thread pool constraints of the HTTP Server (specially the front-end machines), or CPU machine exhaustion, e.g., due to bad code design that causes unnecessary processing. I/O bottlenecks will probably be related to the database (DB) operation, which clearly depend on query complexity, DB configuration and DB access patterns. Network bottlenecks are related to network congestion.

To illustrate this possibility, we propose to systematically collect timing information of one or two web pages from a given server, using the browser side JavaScript Navigation Timing API [24]. Figure 1 depicts the different metrics that are available to this JavaScript library, as defined by the World Wide Web (W3) Consortium. Of these, we will use the most relevant ones for network and server performance: the request time (computed as the time that goes from the request start to the response start) and the response time (which is the time that goes from the response start to the response end). We chose these, because the request and response times are directly related to the request *and* involve server actions, which is not the case of browser processing times, occurring afterwards, or TCP connection times, happening before.

Consider now the following decomposition of the times of interest for us:

- Request Time: client-to-server network transfer time + server processing time + server-to-client network latency.
- Response Time: server-to-client network transfer time.

To make use of these times, we must assume that the server actions, once the server has the first byte of the response ready, do not delay the network transfer of the response. In practice, our analysis depends on the server not causing any delays due to CPU or (disk) I/O, once it starts responding. Note that this is compatible with chunked transfer encoding: the server might compress or sign the next chunk, while delivering the previous one.

We argue that identifying network bottlenecks, and their cause, with time series of these two metrics is actually possible, whenever congestion occurs in both directions of traffic. In this case, the request and response times will correlate strongly. If no network congestion exists, but the response is still slow, the correlation of request and response times will be small, as processing time on the server dominates. Small correlation points to a bottleneck in the server, whereas high correlation points toward the network. Hence, repeated requests to a single resource of the system, such as the entry page can help to identify network congestion, although we cannot tell exactly where in the network does this congestion occur. Henceforth, we will call "single-page request" analysis to this correlation-based evaluation of the request and response time series from a single URL. In this paper, we improve from our previous work [1], by providing evidence in Section V supporting that, when the service is slow, a high (low) correlation between the request and response times results from network (server) congestion.

B. Identification of CPU bottlenecks

Separating CPU from I/O bottlenecks is a much more difficult problem. We resort to a further assumption here: the CPU tasks share a single pool of resources, possibly with several (virtual) machines, while I/O is often partitioned. This, we believe, reflects the conditions of many large systems, as load balancers forward requests to a single pool of machines, whereas data requests may end up in separate DB tables, served by different machines, depending on the items requested. Since scarce CPU resources affect *all* requests, this type of bottleneck synchronizes all the delays (i.e., different parallel requests tend to be simultaneously slow or fast). Thus, logically, unsynchronized delays must point to I/O bottlenecks. On the other hand, one cannot immediately conclude anything, with respect to the type of bottleneck, if the delays are synchronized (requests might be suffering either from CPU or similar I/O limitations).

The challenge is, therefore, to identify pairs of URLs showing unsynchronized delays, to pinpoint I/O bottlenecks. Ensuring that a request for an URL has I/O is usually simple, as most have. In a news site, fetching a specific news item will most likely access I/O. To have a request using only CPU or, at least, using some different I/O resource, one might fetch non-existing resources, preferably using a path outside the logic of the site. We call "independent requests" to this mechanism of using two URLs requesting different types of resources.

One should notice that responses must occupy more than a single TCP [25] segment. Otherwise, one cannot compute any meaningful correlation between request and response times, as this would always be very small.

In our experiments, we will start by evaluating the correlation between request and response times. Then, we will experimentally try the "single-page request" and the "independent

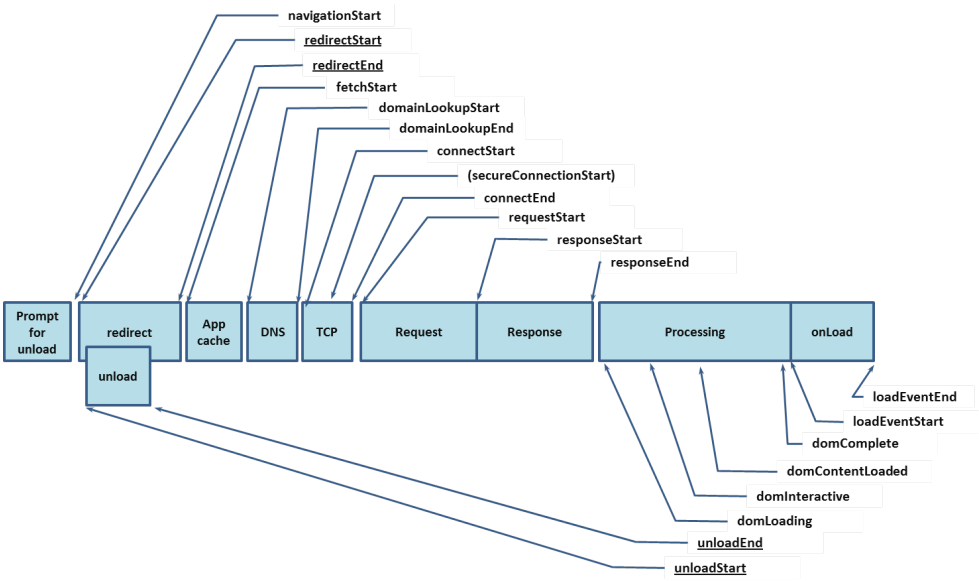


Figure 1. Navigation Timing metrics (figure from [24])

TABLE II. SOFTWARE USED AND DISTRIBUTION.

Component	Observations	Version
Selenium	selenium-server-standalone jar	2.43.0
Firefox	browser	23.0
Xvfb	xorg-server	1.13.3

requests" mechanisms, to observe whether they can actually spot bottlenecks in real web sites.

IV. EXPERIMENTAL EVALUATION

In this section, we present the results of our experimental evaluation. First, we present the setup and afterwards the most important results obtained with the experiments.

A. Experimental Setup

For the sake of doing an online analysis, we used a software testing framework for web applications, called Selenium [26]. The Selenium framework emulates clients accessing web pages using the Firefox browser, thus retaining access to the Javascript Navigation Timing API [24]. We use this API to read the request and response times necessary for the "single-page request" and "independent requests" mechanisms. We used a UNIX client machine, with a crontab process, to request a page each minute [27], using Selenium and the Firefox browser. We emulated a virtual display for the client machine using Xvfb [28]. Table II lists the software and versions used.

One of the criteria we used to choose the pages to monitor was their popularity. However, to conserve space, we only show results of pages that provided interesting results, thus omitting sites that displayed excellent performance during the entire course of the days we tested (e.g., CNN [29] or Amazon [30]) — these latter experiments would have little to show regarding bottlenecks. On the other hand, we could find some bottlenecks in a number of other real web sites:

- **Photo repository** — We kept downloading the same 46 KiloBytes (KiB) Facebook photo, which was actually delivered by a third-party provider Content Delivery Network (CDN). During the time of this test, the CDN was retrieving the photo from Ireland. This experiment displays network performance problems.
- **Portuguese News Site** — this web page is the 5th most used portal in Portugal (only behind Google – domain .pt and .com, Facebook and Youtube) and the 1st page of Portuguese language in Portugal [31]. This web page shows considerable performance perturbations on the server side, especially during the wake up hours.
- **Portuguese Sports News** — This is an online sports newspaper. We downloaded an old 129 KiB news item and an inexistent one for several days. The old news item certainly involves I/O, to retrieve the item from a DB, whereas the inexistent may or may not use I/O, we cannot tell for sure. We ensured a separation of 10 seconds between both requests. One should notice that having a resource URL involving only CPU would be a better choice to separate bottlenecks. However, since we could not find such resource, a non-existing one actually helped us to identify an I/O bottleneck.
- **Social Network Site** — We used the 1st popular social network and the largest social network worldwide. The technology demands are enormous to ensure quality-of-experience to their users and, therefore, preventing bottleneck occurrences. However, recent blackouts in the system have shown the potential of our tool to detect system anomalies and predict web page disruptions.

B. Results

We start by analyzing the results from the Content Delivery Network and from Portuguese News site, in Figures 2, 3, and 4. These figures show the normal behavior of the systems and enable us to identify periods where the response times fall out of the ordinary.

Figure 2 shows the response of the CDN site for a lapse of several days. We can clearly observe a pattern in the

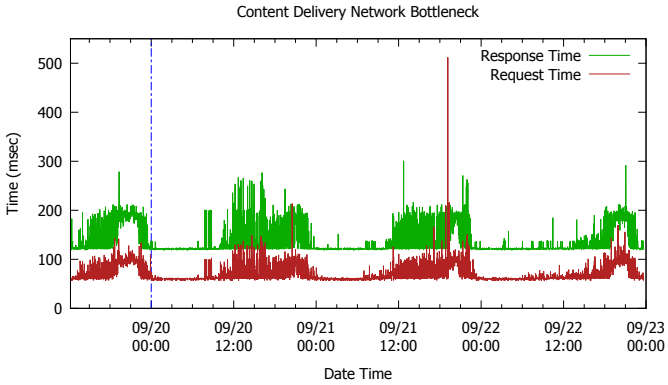


Figure 2. CDN bottleneck.

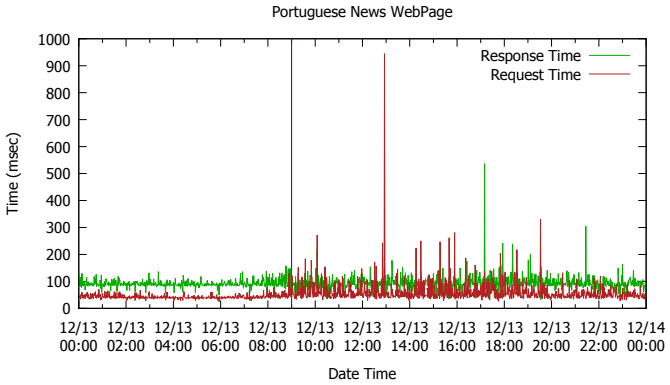


Figure 4. Portuguese News Site bottleneck.

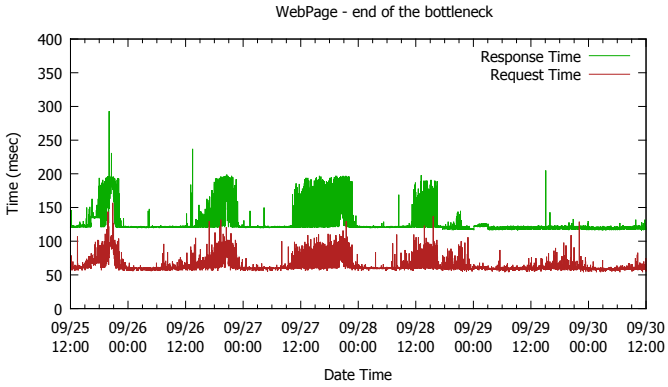


Figure 3. CDN - end of the bottleneck.

response that is directly associated to the hour of the day. During working hours and evening in Europe, we observed a degradation in the request and response times (see, for example, the left area of the blue line on September 19, 2014, a Friday). The green and the red lines (respectively, the response and the request times), clearly follow similar patterns, a sign that they are strongly correlated. Computing the *correlation coefficient* of these variables, $r(Req, Res)$, for the left side of the blue line we have $r(Req, Res) = 0.89881$, this showing that the correlation exists indeed. However, for the period where the platform is more "stable" (between the first peak periods) we have $r(Req, Res) = -0.06728$. In normal conditions the correlation between these two parameters is low. This allows us to conclude that in the former (peak) period we found a network bottleneck that does not exist in the latter. However, our method cannot determine where in the network is the bottleneck. Interestingly, in Figure 3, we can observe that the bottleneck disappeared after a few days. On September 29^{th}, we can no longer see any sign of it.

Regarding Figure 4, which shows request and response times of the main page of a news site, we can make the same analysis for two distinct periods: before and after 9 AM (consider the blue vertical line) of December 13, 2013 (also a Friday). Visually, we can easily see the different profiles of the two areas. Their correlations are:

- $r(Req, Res)_{before9AM} = 0.36621$
- $r(Req, Res)_{after9AM} = 0.08887$

The correlation is low, especially during the peak period, where the response time is more irregular. This case is therefore quite different from the previous one, and suggests that no network bottleneck exists in the system, during periods of intense usage. With the "single-page request" method only, and without having any further data of the site, it is difficult to precisely determine the source of the bottleneck (CPU or I/O).

To separate the CPU from the I/O bottleneck, we need to resort to the "independent requests" approach, which we followed in the Portuguese Sports News case. Figures 5, 6, 7, and 8 show time series starting on February 18^{th}, up to February 21^{st} 2015. We do not show the response times of the inexistent page as these are always 0 or 1, thus having very little information of interest for us. In all these figures, we add a plot of the moving average with a period of 100, as the moving average is extremely helpful to identify tendencies.

Figures 5 and 6 show the request time of the old 129 KiB page request. The former figure shows the actual times we got, whereas in the latter we deleted the highest peaks (those above average), to get a clearer picture of the request times. A daily pattern emerges in these figures, as daytime hours have longer delays in the response than night hours. To exclude the network as a bottleneck, we can visually see that the response times of Figure 7 do not exhibit this pattern, which suggests a low correlation between request and response times (which is indeed low). Next, we observe that the request times of the existent and inexistent pages (refer to Figure 8) are out of sync. The latter seems to have much smaller cycles along the day, although (different) daily patterns seem to exist as well. For the reasons we mentioned before, in Section III, under the assumption that processing bottlenecks would *simultaneously* affect both plots, we conclude that the main source of bottlenecks in the existent page is I/O. This also suggests the impossibility of having the request time dominated by access to a cache on the server, as this would impact processing, thus causing synchronized delays. A final word for the peaks that affect the request time: they weakly correlate with response times. Hence, their source is also likely to be I/O.

Figures 9 and 10 show a period when the social network web page was down in the entire world, due to a system misconfiguration. Figure 9 shows how the page behaved, regarding request and response times — before, during and

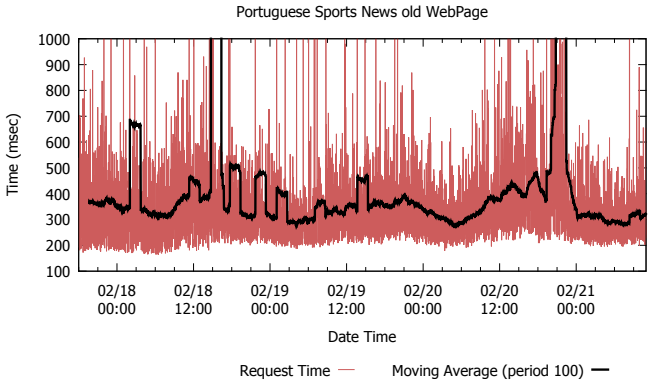


Figure 5. Portuguese Sports News old page — request times.

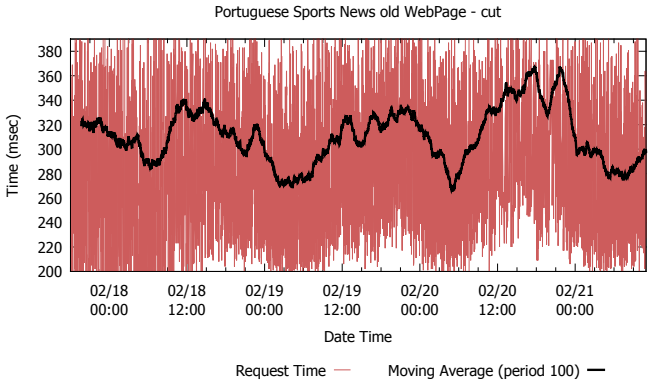


Figure 6. Portuguese Sports News old page — request times with peaks cut.

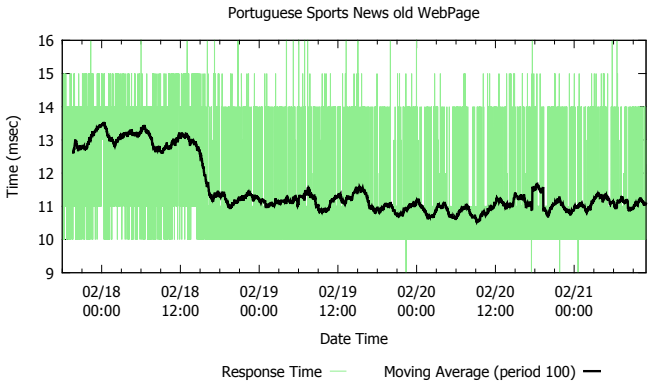


Figure 7. Portuguese Sports News old page — response times.

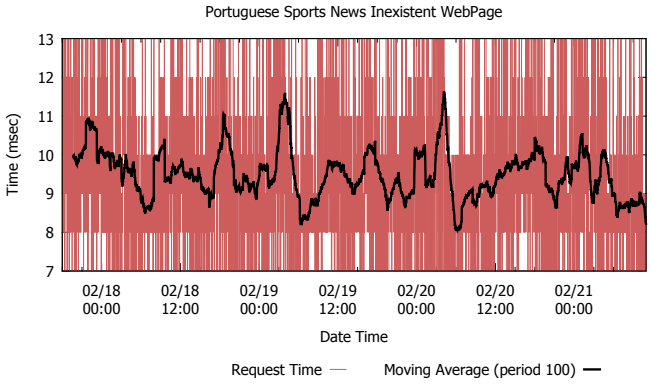


Figure 8. Portuguese Sports News inexistent page — request times.

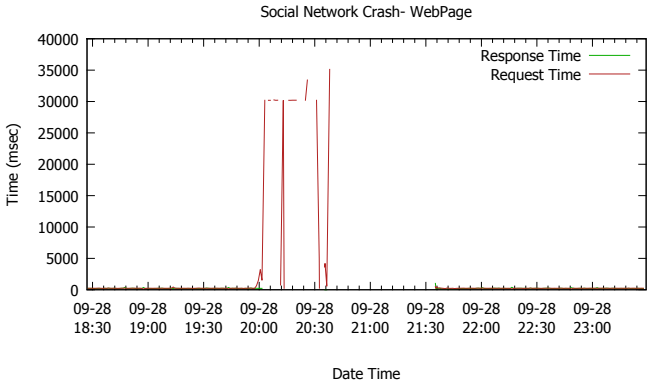


Figure 9. Social Network Web Page crash.

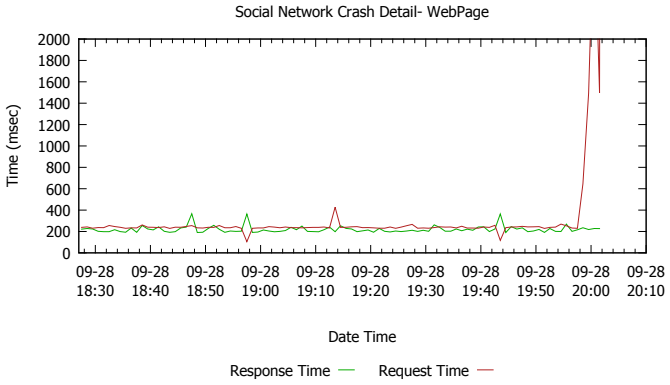


Figure 10. Social Network Web Page crash detail.

after the system resumed responding correctly. Figure 10 gives a closer look of the period before the web page failure. Time periods without request or response times, occurred when the client reached the configured timeout and aborted the web page request. Currently, the timeout is configured to be 60 seconds. Analyzing Figure 9, we can identify the period of time when the web page was down or responding incorrectly. This might be important, if the web page is hosted in a third-party provider that might be held responsible for the failure and the user wants to complain for a refund [32–34]. Figure 10 gives a closer look of the minutes before the failure. The request time increased significantly, while the response time remained unchanged — a weak correlation that points to a server-side bottleneck. This agrees with what was mentioned by the media [35]. Additionally, prior to the complete failure, we can observe a 5-minute window, were the problem was starting to become apparent, and that could be used to fix the problem or alert system administrators.

V. Client-Side Monitoring Using Two Distinct Networks

In Section III, we focused on the problem of detecting bottlenecks using only client metrics. We used a time series approach based on the assumption that we can distinguish server from network congestions, by looking at the correlation between the request and response times. Unfortunately, since we have no access to the internal server-side data of the web servers we monitored, we cannot directly confirm this

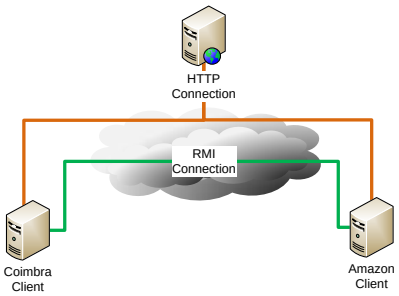


Figure 11. Experimental Setup - 2 clients

correlation hypothesis. To confirm this hypothesis and thus verify the feasibility of this method, we now resort to two distinct clients. The idea is straightforward: the observation from two different clients should be coherent; clients should agree on whether a bottleneck exists, and where does it come from. As we shall see, the results we got slightly deviated from what we expected, thus raising a very interesting question.

A. Experimental Settings

We used the same technologies mentioned in Table II for each client. This means that each client was running a Selenium instance to invoke a specific web page, through Firefox. However, having two unsynchronized clients would invalidate the results, because we would not know if the requests were made at the same time. To eliminate this limitation we created a communication protocol between the clients in Java RMI [36]. Before fetching a page, the two clients communicate with each other, to determine which page to get and to ensure some degree of synchronicity. The process consists in 3 steps — first, client A notifies client B to invoke a determined URL; second, both clients invoke the URL and save the request and response time; and finally, client A receives the data from the web page invocation from client B. One of the clients (client A in this description) has the role of "master", triggering the web page invocation and the collection data from both requests. Clients should be in different locations to have different network connectives. We picked our own department facilities in Coimbra, and a virtual machine in the Amazon Web Service cloud in the Northern Virginia Region [37]. Figure 11 illustrates this interaction — the RMI connection is identified in green, whereas the HTTP connections to the web page are depicted in orange.

One of the criteria we used to choose the pages to monitor was their popularity. Additionally, the web page should have the same location regardless of the origin of the client (Amazon in America or Coimbra in Europe). To ensure this, we compared the IP given by the DNS to each client, to ensure that they were, indeed, monitoring the same server. A second criterion was to monitor web pages from different geographic locations. However, to conserve space, we only show results of pages that provided interesting results. Among these, we could find some bottlenecks in the following real web sites:

- **American electronic commerce and cloud computing company** — We kept downloading the main page from this popular web page hosted in the United States from our two clients. This experiment displays a significant performance improvement, at some point in time. This improvement was observed in both clients.
- **Chinese Search Engine** — this web page is the front-end for one of the most popular search engines in the world [31]. It is hosted in China. This web page shows some network perturbations during a specific time in both clients.
- **Portuguese Sports News** — This is an online sports newspaper already used in Section III. The web page is located in Portugal (Europe). We downloaded the main page during several days. We verified several pattern changes associated with system bottlenecks in both clients.

B. Results

Since we now have two clients invoking the same URL, at the same time, we expect similar server response patterns at both clients. One would assume that whenever the response time pattern in only one of the clients changes, the difference should result from some bottleneck in the client-server network path that is specific to the client observing the change. However, if both clients observe a modification in the response patterns (in terms of request and response time series), we can conclude that this is the result of a component that is common to both clients. Hence, this is the result of a system bottleneck or a common network path.

We will now experimentally try the two clients "synchronous request" mechanism, to observe whether they can actually spot bottlenecks in real web sites and achieve consistent observation of response patterns.

We start by analyzing the results of the American electronic commerce web page in Figure 12, which shows the response of the main page for a lapse of several days. The pair of figures mostly shows normal behavior seen in Coimbra and in the AWS, thus allowing us to identify periods when the response times fell out of the ordinary for one or both clients. We can clearly observe a pattern in the response that is directly associated to the hour of the day. Additionally, the pattern exists for both clients, this meaning that both were experiencing the same constraints (system or network) from the web page. To have a better understanding of the trends, we also show a moving average of the last 100 samples of response time, in black. Computing the *correlation coefficient* for the response and request times for both clients, $r(Req, Res)$, for the interval between September 12^{th} 13:00 and September 13^{th} 13:00 2015, we get a correlation of $r(Req, Res)_{Coimbra} = 0.13896$ and $r(Req, Res)_{AWS} = -0.07370$. Since none of the clients observed significant congestion conditions, these low correlations provide us very little information and suggest a normal behavior of the system. Near the end of the experiment, still in Figure 12, we can see an improvement in the response times of both clients. Calculating the *correlation coefficient* for this period, we have $r(Req, Res)_{Coimbra} = 0.07974$ and $r(Req, Res)_{AWS} = 0.14808$. Hence, having in consideration the low correlations and what was mentioned in Section III, we can infer that the improvement experienced by both clients seems to be a consequence of a change in the American electronic commerce web page. An improvement in the system network is less likely, because the request time rested unchanged for this period.

Figure 13 shows the request and response times of the Chinese Search Engine web page. The web page presents a relatively stable pattern during most of the days. During

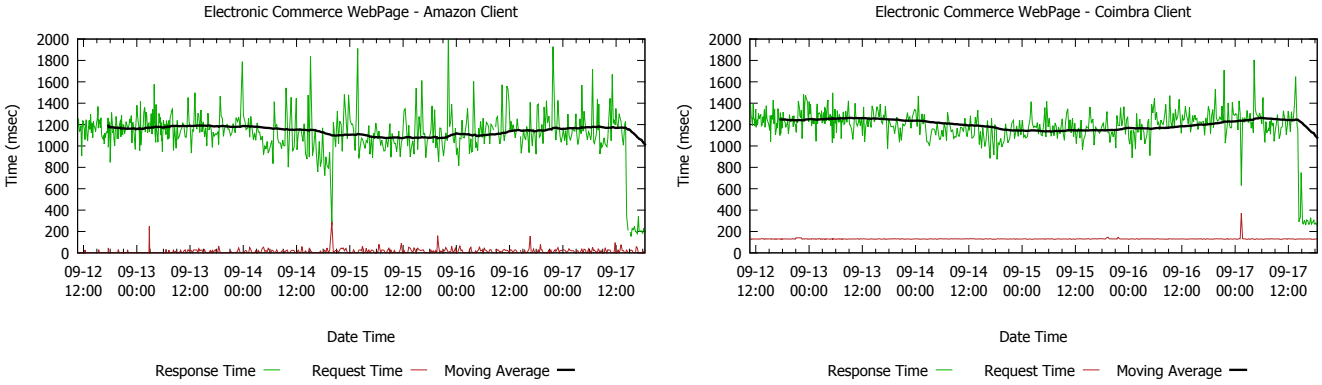


Figure 12. American electronic commerce web page

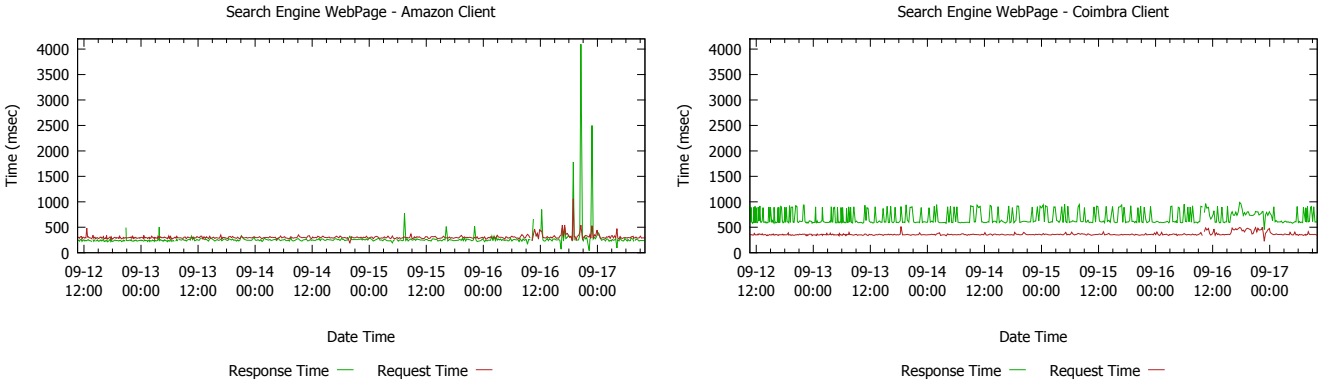


Figure 13. Chinese search engine web page

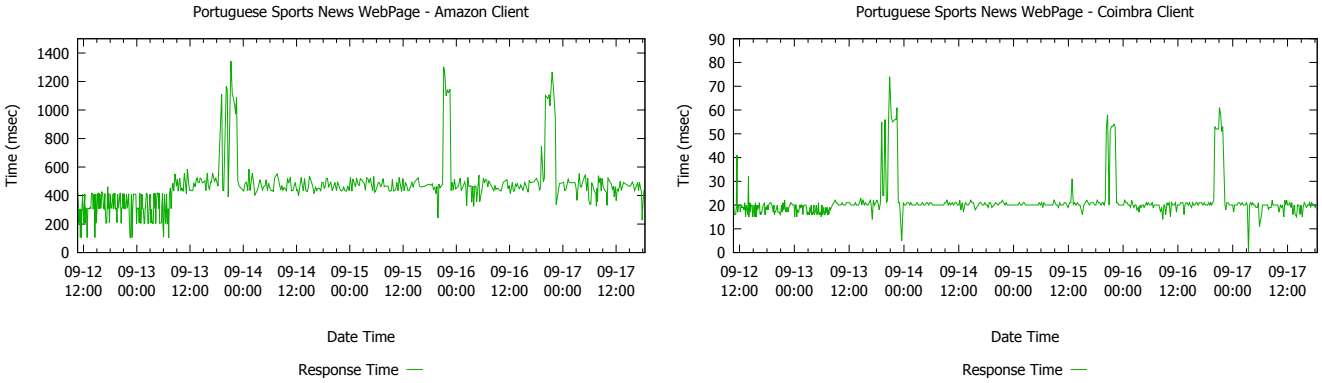


Figure 14. Portuguese sports news web page

this period (before September 16^{th}), the *correlation coefficient* was $r(Req, Res)_{Coimbra} = 0.04532$ and $r(Req, Res)_{AWS} = 0.16566$, this meaning that in normal conditions there was no correlation between request and response times. However, for the period after September 16^{th}, there was a significant change observed by the AWS client, and although less significant, also by the Coimbra client. This is even clearer when we calculate the *correlation coefficient* of both clients for this period. The correlation in Coimbra was $r(Req, Res)_{Coimbra} = 0.69707$ and in AWS $r(Req, Res)_{Aws} = 0.57794$. This means that the correlation for request and response times in both clients increased significantly, when compared to the normal pattern. Hence, taking in consideration what was mentioned in Section III, we are, most likely, observing a network bottleneck in a common path between the server and the clients.

Figure 14 shows a degradation of the response time in 3 distinct moments. This degradation was observed in both clients at the same time. When we calculate the correlation

TABLE III. CORRELATION FOR THE 3 PEAKS OF PORTUGUESE SPORTS NEWS SITE

Correlation	Coimbra	AWS
$r(Req, Res)_{1^{st}peak}$	-0.32036	0.35263
$r(Req, Res)_{2^{nd}peak}$	0.33789	0.35815
$r(Req, Res)_{3^{rd}peak}$	0.28955	0.15749

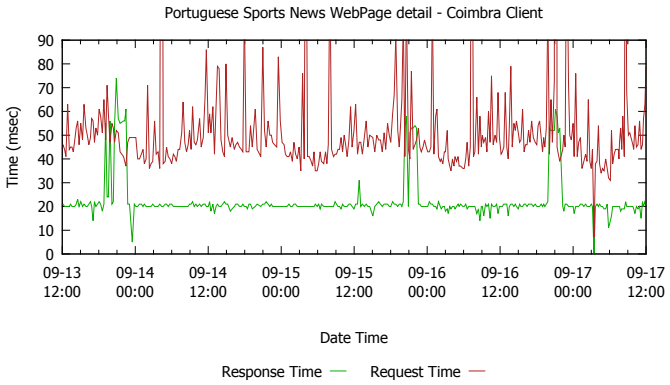


Figure 15. Portuguese Sports News Web Page - Detail

for a stable period of good performance (e.g., between the first and second peak), we get $r(Req, Res)_{Coimbra} = -0.02381$ in Coimbra and $r(Req, Res)_{AWS} = 0.17699$ in the AWS. Then, for the three observed peaks, we have the correlation values of Table III.

This *correlation coefficient* for both clients in the three peaks differs considerably, especially in Coimbra. We show with finer detail the request and response times of this client in Figure 15. The correlation is never very high, thus pointing to a server problem, especially in the first and third peaks. However, it is also never low, thus suggesting that the network, more specifically a common path of the network might have been affected as well, at least in the second peak.

VI. AUTOMATIC DETECTION OF BOTTLENECKS USING TWO DISTINCT NETWORKS

Our next step was to do a simple automated mechanism to detect bottlenecks, using information coming from the pair of clients. This is beneficial, not only because automation may eventually lead to a quicker detection of performance problems, but mainly because a visual inspection as we did in the previous sections is error-prone and is subject to all sort of biases. Indeed, with this new scheme we were able to achieve new conclusions and get a deeper insight of performance bottleneck problems.

A. Overview

The pair of clients provides four different variables that we can feed into an algorithm: a boolean value per client telling whether or not the client sees a congestion; and the correlation between the request and the response times, for both clients. Determining if the client is observing a congestion in the service is not trivial, in the sense that different algorithms may respond differently. In our experiments, we used an algorithm based on a moving average. Unlike congestion, the correlation is easier to determine. As before, we use the last 100 metrics

TABLE IV. All possible combinations of congestion and correlation

Client 1		Client 2		Cause
Congestion?	Correlation	Congestion?	Correlation	
No	Irrelevant	No	Irrelevant	No Bottleneck
No	Irrelevant	Yes	Low	Impossible
No	Irrelevant	Yes	High	Client 2's Network
Yes	Low	Yes	Low	Server
Yes	Low	Yes	High	Server & C. 2 Net.
Yes	High	Yes	High	Common Network

of request and response times. Note that for the correlation to tell us something, we must be careful enough to request pages that are relatively large, but still go in a single non-chunked HTTP message. Considering high and low correlations, we get 16 possible combinations of the four variables, of which we arrange the cases of interest in Table IV. We omit the redundant cases, where client 1 and client 2 would simply swap their variables. For example, since we already have "Yes, Low, Yes, High", in the line before last, it would be pointless to include a "Yes, High, Yes, Low".

Line 1 of the table is pretty much trivial: none of the clients observes a bottleneck, therefore, looking for correlation is not relevant. In line 2, one of the nodes observes a congestion that does not come from the network (low correlation). Hence, the other node should also observe a congestion. Hence, this line is seemingly impossible. However, as we shall discuss, it may, in fact, happen. In line 3, the client observing the congestion can tell from the correlation that the network is congested. Since client 1 observes no congestion, the network of client 2 is the culprit. The following case is equally straightforward: when both clients observe a congestion and a low correlation, the server is the culprit. In the following case, more than one bottleneck exists: client 1 can tell that the server is the most likely cause for the low correlation between request and response times, because responses are taking quite a long time. On the other hand, client 2 is also observing congestion, but it can see a high correlation in the request and response times, thus concluding that the problem lies in the network. Despite seeming contradictory, this is possible, if both the server and client 2 network are sources for delays. In the final line, the problem lies in the network that is common to clients 1 and 2. We will now try to confirm to what extent do real observations actually fit into this model.

B. Algorithm to detect bottlenecks

In this section, we present an algorithm that evaluates the variables of Table IV and outputs the cause of the problem. The algorithm combines the request and response time series retrieved at the same time, by two distinct clients for the same web page, collected in our "synchronous request" experiments. We wrote the algorithm in Python and we describe its high-level details in pseudo-code in Algorithm 1. The expression CongestionClient (1 or 2) $\geq$ CongestionThreshold becomes true when the moving average of the request plus receive time of the last 100 values grows above 15% of the average of all samples, for that client. We consider the threshold that splits low from high correlation to be 0.2. The correlation also takes into account the last 100 measurements.

C. Results

In this section, we present some of the results that we obtained with Algorithm 1 and compare them to the data previously analyzed in Section V. Although we ran our algorithm

Algorithm 1 Identify Bottleneck

```
if CongestionClient1 ≥CongestionThreshold then
  if CongestionClient2 ≥CongestionThreshold then
    if CorrelClient1 ≥HighCorrelationThreshold then
      if CorrelClient2 ≥HighCorrelationThreshold then
        Common Network
      else
        Client 1's Network and Server
      end if
    else if CorrelClient2 ≥ HighCorrelationThreshold
    then
      Client 2's Network and Server
    else
      Server
    end if
  else
    if CorrelClient1 ≥ HighCorrelationThreshold then
      Client 1's Network
    else
      Impossible
    end if
  end if
else if CongestionClient2 ≥ CongestionThreshold then
  if CorrelClient2 ≥HighCorrelationThreshold then
    Client 2's Network
  else
    Impossible
  end if
else
  No Bottleneck
end if
```

under varied conditions, we will focus on its responses for the inputs depicted in Figures 12, 13, and 14. We can say that all the bottlenecks we identified by visual inspection in these figures were also identified by the algorithm, which pointed out the same sources for problems. Although this might result from the specific thresholds we selected and from the size of the sliding window in the moving average and network correlation, the effort of tuning the algorithm and making it run under real data allowed us to reach two results:

- The algorithm cannot cope with request or response times that are too low. For instance, if the request or response times fall to values near the millisecond range, as in one case where the client and server were very close to each other, any small increase in the response time, no matter how small it is, will look as a congestion.
- The algorithm tags some congestion cases as being impossible (line 2 in Table IV). This happens because some requests take so long to get an answer that they are able to push the moving average above the congestion threshold. The interesting thing is that this sort of delay, which happens rarely, is usually not seen by the peer client, which is fetching pages from the same server; and it is not seen neither before, nor after, by the same client. This suggests that some requests get an unfair amount of wait. If this is really the case, what is the source of the problem (network, CPU, or I/O) and how often does this happen remains as an open question.

VII. Discussion and Conclusion

We proposed to detect bottlenecks on HTTP servers using client-side observations of request and response times. A comparison of these signals, either over the same, or a small number of resources, enables the identification of CPU, network and I/O bottlenecks. We did this work having no access to internal server data and mostly resorting to visual inspection of the request and response times. If run by the owners of the site, we see a number of additional options:

- Simply follow our approach of periodically, invoking URLs in one or more clients, as a means to complement current server-side monitoring tools. This may help to reply to questions such as "what is the impact of a CPU occupation of 80% for interactivity?".
- A hybrid approach, with client-side and server-side data is also possible. I.e., the server may add some internal data to *each* request, like the time the request takes on the CPU or waiting for the database. Although much more elaborate and dependent on the architecture, instrumenting the client and the server sides is, indeed, the only way to achieve a full decomposition of request timings.
- To improve the quality of the analysis we did in Section IV, site owners could also add a number of very specific resources, like a page that has known access time to the DB, or known computation time.
- It is also possible to automatically collect timing information from real user browsers, as in Google Analytics [38], to do subsequent analysis of the system performance. In other words, instead of setting up clients for monitoring, site owners might use their real clients, with the help of some JavaScript.

In summary, we collected evidence in support of the idea of identifying bottlenecks from the client side. In our previous work [1], we recognized that to unambiguously demonstrate these results we needed further evidence from a larger number of sites, and from supplementary server data. We now managed to run several experiments with a second client. Although mostly concurring with our initial observations, the second client opened an entirely new perspective: sometimes one of the clients observes a delay in a single very concrete request, which is neither observed by the other client, nor by the client itself, either before or after. I.e., even when the server seems to be delivering a normal service, clients may occasionally fail to receive a response in reasonable time. We are certain that the problem does not come from the network, because we classify network problems in a different category. This result agrees with [1]. We thus know that the server itself is the culprit for such delays.

Let us think for a moment on the unique route taken by each request: the TCP connection takes the request up to the server, where some thread reads it, processes it, and (most likely) forwards it to another layer of the system, where some thread will eventually fetch several items from the database, before enqueuing or sending the response back to the client. Each request might follow slightly different routes, depending on the threads that get it. This suggests a simple, but significant conclusion: a few unlucky requests get blocked at some point

inside the server. To be fair, there was never any guarantee that *all* requests would get their fair chance, or that they would all get a quick response. But observing such cases in a moderate number of samples is, we think, a rather interesting result. This observation raises the question of determining the exact mechanism behind starvation of some specific requests, and how likely is such mechanism to come into play.

REFERENCES

[1] R. Filipe, S. Boychenko, and F. Araujo, "Online client-side bottleneck identification on HTTP server infrastructures," in *The Tenth International Conference on Internet and Web Applications and Services (ICIW 2015)*, Brussels, Belgium, June 2015, pp. 22–27.

[2] *RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1*, Internet Engineering Task Force (IETF), Internet Engineering Task Force (IETF) Std., June 1999. [Online]. Available: http://www.faqs.org/rfcs/rfc2616.html

[3] D. Battre, M. Hovestadt, B. Lohrmann, A. Stanik, and D. Warneke, "Detecting bottlenecks in parallel dag-based data flow programs," in *Many-Task Computing on Grids and Supercomputers (MTAGS), 2010 IEEE Workshop on*, 2010, pp. 1–10.

[4] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Adaptive resource provisioning for read intensive multi-tier applications in the cloud," *Future Generation Computer Systems*, vol. 27, no. 6, pp. 871–879, 2011.

[5] Y. Shoaib and O. Das, "Using layered bottlenecks for virtual machine provisioning in the clouds," in *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, 2012, pp. 109–116.

[6] N. Huber, F. Brosig, and S. Kounev, "Model-based self-adaptive resource allocation in virtualized environments," in *Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, ser. SEAMS '11. New York, NY, USA: ACM, 2011, pp. 90–99. [Online]. Available: http://doi.acm.org/10.1145/1988008.1988021

[7] P. Bodík, R. Griffith, C. Sutton, A. Fox, M. Jordan, and D. Patterson, "Statistical machine learning makes automatic control practical for Internet datacenters," in *Proceedings of the 2009 conference on Hot topics in cloud computing*, ser. HotCloud'09. Berkeley, CA, USA: USENIX Association, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1855533.1855545

[8] Q. Wang, Y. Kanemasa, J. Li, D. Jayasinghe, T. Shimizu, M. Matsubara, M. Kawaba, and C. Pu, "Detecting transient bottlenecks in n-tier applications through fine-grained analysis," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, July 2013, pp. 31–40.

[9] Q. Zhang, L. Cherkasova, and E. Smirni, "A regression-based analytic model for dynamic resource provisioning of multi-tier applications," in *Autonomic Computing, 2007. ICAC '07. Fourth International Conference on*, June 2007, pp. 27–27.

[10] G. Franks, D. Petriu, M. Woodside, J. Xu, and P. Tregunno, "Layered bottlenecks and their mitigation," in *Quantitative Evaluation of Systems, 2006. QEST 2006. Third International Conference on*, Sept 2006, pp. 103–114.

[11] S. Malkowski, M. Hedwig, J. Parekh, C. Pu, and A. Sahai, "Bottleneck detection using statistical intervention analysis," in *Managing Virtualization of Networks and Services*. Springer, 2007, pp. 122–134.

[12] S. Malkowski, M. Hedwig, and C. Pu, "Experimental evaluation of n-tier systems: Observation and analysis of multi-bottlenecks," in *Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on*. IEEE, 2009, pp. 118–127.

[13] R. Chi, Z. Qian, and S. Lu, "A heuristic approach for scalability of multi-tiers web application in clouds," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2011 Fifth International Conference on*, 2011, pp. 28–35.

[14] R. Singh, U. Sharma, E. Cecchet, and P. Shenoy, "Autonomic mix-aware provisioning for non-stationary data center workloads," in *Proceedings of the 7th international conference on Autonomic computing*, ser. ICAC '10. New York, NY, USA: ACM, 2010, pp. 21–30. [Online]. Available: http://doi.acm.org/10.1145/1809049.1809053

[15] S. Agarwal, N. Liogkas, P. Mohan, and V. Padmanabhan, "Webprofiler: Cooperative diagnosis of web failures," in *Communication Systems and Networks (COMSNETS), 2010 Second International Conference on*, Jan 2010, pp. 1–11.

[16] W. Iqbal, M. N. Dailey, D. Carrera, and P. Janecek, "Sla-driven automatic bottleneck detection and resolution for read intensive multi-tier applications hosted on a cloud," in *Advances in Grid and Pervasive Computing*. Springer, 2010, pp. 37–46.

[17] H. Liu and S. Wee, "Web server farm in the cloud: Performance evaluation and dynamic architecture," in *Proceedings of the 1st International Conference on Cloud Computing*, ser. CloudCom '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 369–380. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-10665-1_34

[18] B. Singh and P. Nain, "Article: Bottleneck occurrence in cloud computing," *IJCA Proceedings on National Conference on Advances in Computer Science and Applications (NCACSA 2012)*, vol. NCACSA, no. 5, pp. 1–4, May 2012, published by Foundation of Computer Science, New York, USA.

[19] R. Filipe, S. Boychenko, and F. Araujo, "On client-side bottleneck identification in HTTP servers," in *Proceedings of the 5th INForum — Simpósio de Informática*, Covilh, Portugal, September 2015.

[20] M. Attariyan, M. Chow, and J. Flinn, "X-ray: automating root-cause diagnosis of performance anomalies in production software," in *Proceedings of the 10th USENIX conference on Operating Systems Design and Implementation*, ser. OSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 307–320. [Online]. Available: http://dl.acm.org/citation.cfm?id=2387880.2387910

[21] F. Oliveira, A. Tjang, R. Bianchini, R. P. Martin, and T. D. Nguyen, "Barricade: defending systems against operator mistakes," in *Proceedings of the 5th European conference on Computer systems*, ser. EuroSys '10. New York, NY, USA: ACM, 2010, pp. 83–96. [Online]. Available: http://doi.acm.org/10.1145/1755913.1755924

[22] "Papers — HP Web server performance tool," http://www.hpl.hp.com/research/linux/httperf/, retrieved: May, 2015.

[23] "Performance tools — Apache JMeter^{TM}," http://jmeter.apache.org/, retrieved: May, 2015.

[24] "Papers — Navigation Timing," https://dvcs.w3.org/hg/webperf/raw-file/tip/specs/NavigationTiming/Overview.html, retrieved: May, 2015.

[25] J. Postel, "Transmission Control Protocol," RFC 793 (Standard), Internet Engineering Task Force, Sep. 1981, updated by RFCs 1122, 3168. [Online]. Available: http://www.ietf.org/rfc/rfc793.txt

[26] "Papers — Selenium Browser automation," http://www.seleniumhq.org/, retrieved: May, 2015.

[27] "Crontab - quick reference — admin's choice - choice of unix and linux administrators," http://www.adminschoice.com/crontab-quick-reference, retrieved: May, 2015.

[28] "Xvfb," http://www.x.org/archive/X11R7.6/doc/man/man1/Xvfb.1.xhtml, retrieved: May, 2015.

[29] "Breaking news, u.s., world, weather, entertainment & video news - cnn.com," http://edition.cnn.com, retrieved: May, 2015.

[30] "Amazon.com: Online shopping for electronics, apparel, computers, books, dvds & more," http://www.amazon.com, retrieved: May, 2015.

[31] "Alexa — Top Sites in Portugal," http://www.alexa.com/topsites/countries/PT, retrieved: May, 2015.

[32] "Papers — Windows Azure Service Level Agreement," http://www.windowsazure.com/en-us/support/legal/sla/, retrieved: May, 2015.

[33] "Papers — HP Service Level Agreement," https://www.hpcloud.com/SLA, retrieved: May, 2015.

[34] "Papers — Amazon EC2 Service Level Agreement," http://aws.amazon.com/ec2-sla/, retrieved: May, 2015.

[35] "Facebook crash," http://www.dailymail.co.uk/sciencetech/article-3252603/Facebook-goes-Social-network-crashes-time-month-leaving-users-panic.html, retrieved: Nov, 2015.

[36] "Papers — RMI overview," https://docs.oracle.com/javase/tutorial/rmi/overview.html, retrieved: Nov, 2015.

[37] "Papers — Amazon Web Services," http://aws.amazon.com/, retrieved: Nov, 2015.

[38] B. Clifton, *Advanced Web Metrics with Google Analytics*. Alameda, CA, USA: SYBEX Inc., 2008.

Business Process Risk Management and Simulation Modelling for Digital Audio-Visual Media Preservation

Vegard Engen, Galina Veres, Simon Crowle and Paul Walland
IT Innovation Centre, University of Southampton
Southampton, United Kingdom
Email: {ve, gvv, sgc, pww}@it-innovation.soton.ac.uk

Christoph Bauer
Multimedia Archives, Austrian Broadcasting Corporation
Vienna, Austria
Email: christoph.bauer@orf.at

***Abstract*—Digitised and born-digital Audio-Visual (AV) content presents new challenges for preservation and Quality Assurance (QA) to ensure that cultural heritage is accessible for the long term. Digital archives have developed strategies for avoiding, mitigating and recovering from digital AV loss using IT-based systems, involving QA tools before ingesting files into the archive and utilising file-based replication to repair files that may be damaged while in the archive. However, while existing strategies are effective for addressing issues related to media degradation, issues such as format obsolescence and failures in processes and people pose significant risk to the long-term value of digital AV content. We present a Business Process Risk management framework (BPRisk) designed to support preservation experts in managing risks to long-term digital media preservation. This framework combines workflow and risk specification within a single risk management process designed to support continual improvement of workflows. A semantic model has been developed that allows the framework to incorporate expert knowledge from both preservation and security experts in order to intelligently aid workflow designers in creating and optimising workflows. The framework also provides workflow simulation functionality, allowing users to a) understand the key vulnerabilities in the workflows, b) target investments to address those vulnerabilities, and c) minimise the economic consequences of risks. The application of the BPRisk framework is demonstrated on a use case with the Austrian Broadcasting Corporation (ORF), discussing simulation results and an evaluation against the outcomes of executing the planned workflow.**

Keywords–Risk management; business processes; workflows; semantic modelling; simulation modelling.

I. INTRODUCTION

Digital preservation aims to ensure that cultural heritage is accessible for the long term. From the 20th century onwards, AV content has provided a significant record of cultural heritage, and increasing volumes of AV content that have been digitised from analogue sources or produced digitally present new preservation challenges. The focus is no longer on reducing damage to the physical carrier by maintaining a suitable environment; rather, archives must ensure that the significant characteristics of the content, represented digitally, are not lost over time. Digital data enables easier transfer, copying, processing and manipulation of AV content, which is at once a boon but also a problem that requires continuous and active data management.

Digital damage is defined here as any degradation of the value of the AV content with respect to its intended use by a designated community that arises from the process of ingesting, storing, migrating, transferring or accessing the content. The focus here is on strategies that can be used to minimise the risk of loss; e.g., loss of data files or information such as metadata required to view or process the data. In particular, we focus on dealing with issues resulting from system errors, rather than random failure or corruption, considering the risks to the AV content as it is being manipulated by various activities in a workflow process. This includes risks introduced by the people, systems and processes put in place to keep the content safe in the first place.

Archival processes dealing with digital AV content are underpinned by IT systems. In the few years that archives have been working with digitised and born-digital content, best practice in terms of digital content management has rapidly evolved. Strategies for avoiding, reducing and recovering from digital damage have been developed and focus on improving the robustness of technology, people and processes. These include strategies to maintain integrity, improve format resilience and interoperability, and to combat format obsolescence. We will return to this in the following sections.

This paper builds on [1], presenting the research and development work of a Business Process Risk management framework (BPRisk) developed in the EC FP7 DAVID project [2], which combines risk management with workflow specification. BPRisk has been designed to support a best practice approach to risk management of digital AV processes (and thus the content itself). In this paper, we will give an overview of this framework, focusing on semantic modelling, risk specification and simulation modelling. Within the DAVID project, this research and development has been conducted to provide a tool to help prevent damage to digital AV content in broadcasting archives, although the approach is clearly applicable to any digital archive management process where the same challenges of workflow and migration risk are present.

The BPRisk framework is generic in nature, supporting risk specification for Business Process Modelling Notation (BPMN) 2.0 [3] workflows in any domain. The framework utilises a novel semantic risk model developed in the project that encapsulates domain knowledge generated in the DAVID project on known risks (and controls) associated with activities in a controlled vocabulary for the domain of digital preservation (also developed in the project). This enables the framework to be an effective support tool to users who

are typically not familiar with formal risk management. The semantic risk modelling provides the domain experts with a starting point for conducting risk analysis, and semantic reasoning is utilised to provide suggestions of relevant risks and controls for the activities in the respective workflows at design time.

Another focus of this paper is the simulation modelling adopted in the BPRisk framework. We have developed models in order to simulate the execution of workflows, accounting for risk occurrences and their associated impact (e.g., damage to data that renders files unreadable). The purpose of the simulation modelling is to help an organisation reduce costs by designing or optimising workflows in order to reduce the likelihood or impact of risks occurring. For example, it could be used to help justify expenses on technology and control tools, showing the anticipated cost of dealing with issues (risks) when they are not addressed (controlled) versus the cost of preventing them. That is, if the cost of prevention is less, one could argue an anticipated Return On Investment (ROI). Moreover, the simulations can help identify the key vulnerabilities in a workflow in order to help target investments. The aim is to expose issues at design-time so that workflow designers can address them before a workflow is actually executed.

In the DAVID project, the risk management work presented in this paper is one of the four cornerstones of interlinked work on i) understanding damage (how it occurs and its impact), ii) detecting and repairing damage, iii) improving the quality of digital AV content, and iv) preventing damage to digital AV content and ensuring its long-term preservation. The latter is a significant challenge, despite the advances in i) to iii), especially with respect to format obsolescence and failure in processes and people who handle the digital content, which is discussed further below.

The challenges and related work on digital preservation are discussed in Section II. Risk management in this domain is discussed in Section III. Thereafter, in Section IV, we present the BPRisk framework that has been developed in the DAVID project. Following this, we present and discuss further details of the semantic risk modelling and simulation modelling adopted in the framework in Sections V and VI, respectively. Section VII discusses the application of BPRisk on a real use case with the Austrian Broadcasting Corporation. This includes simulation results from the planning stage of the workflow development, and a comparison with the outcomes from executing the workflow. Section VIII concludes this paper and discusses future work.

II. Digital Preservation

AV content is generated in vast quantities from different sources such as film, television and online media, environmental monitoring, corporate training, surveillance and call recording. There are many reasons why content needs to be retained and archived, which might be to enable content re-use for commercial, educational or historical purposes, but equally it might need to be retained and accessible due to regulatory compliance, for security or recording health and safety issues.

Historically, the preservation of analogue content has been intrinsically linked to its method of production; specifically, the media that is used to carry the signal (the carrier). This means that archives preserved 'masters' on magnetic tape, film and even phonograph cylinders [4]. Where masters no longer exist or content was not professionally produced, archives needed to preserve 'access' copies on media such as vinyl records, VHS/Betamax tapes, and audio cassettes. To reduce the risk of damage, archives had to consider the physical characteristics of the media and care for the physical environment to which the media was sensitive (e.g., light, heat, humidity and dust) and to look after the machines that read the media. To increase the chances of being able to read the content again, archives often created copies of the artefact, in case one copy was damaged.

Nowadays, AV content is commonly born-digital and archives such as INA (the French national archive) and ORF (the Austrian broadcaster), who were partners in the DAVID project, have initiated digital migration projects to digitise the older, analogue, content [5]. Digital content (digitised or born digital) can be copied, transferred, shared and manipulated far more readily than its analogue equivalent. In a world of digital AV content, preservation is largely agnostic to the carrier that is used to store and deliver the content. Therefore, preservation and archiving is about making sure that the digital data is safe and that processes that manipulate the data do not cause damage. When referring to 'digital damage' in this paper, it is worth noting the following definition:

> *"Digital damage is any degradation of the value of the AV content with respect to its intended use by a designated community that arises from the process of ingesting, storing, migrating, transferring or accessing the content."* [5]

The above definition is broad, covering damage arising from failure of equipment used to store and process digital content, as well as that arising from human error or from 'failure' of the preservation process. The challenge for digital preservation is to keep the AV content usable for the long-term, which is threatened by format obsolescence, media degradation, and failures in the very people, processes and systems designed to keep the content safe and accessible [6], [7], [8].

Therefore, the core problem is greater than the potential for a digital file already in the archive to become damaged over time due to, e.g., bit rot [6], which can effectively be addressed by keeping multiple copies of each file [5], [7]. We also need to consider the future challenges for digital preservation as some analyses [9] predict that as more 8K AV content is ingested into archives, the growth in data volumes may outstrip the predicted growth in data capacity. More importantly still, the data write speed necessary to store these high data volumes at real time will not be achievable, meaning that it will become impossible to cost-effectively store and replicate such content as it is produced. Therefore, strategies such as file-level replication may not be feasible in the future, and managing risk to the entire workflow process, and determining the most cost-effective archive management approach becomes essential.

III. Risk Management for Digital Preservation

Risk management, in a broad sense, can be understood as *"the coordinated activities to direct and control an organisation with respect to risk"* [10]. Risk, as defined by ISO 31000 [10], is the *"effect of uncertainty on objectives"*. In this context, *uncertainty* arises from random or systematic failure of preservation systems and processes (that may involve manual human activities). The *effect* of which is to cause damage to AV content. In general terms, we can say that the

key *objective* is to ensure long-term preservation of digital AV content, i.e., avoid damage and ensure that it can be accessed in the future.

Current archives such as the French national archive (INA) and the Austrian Broadcasting Corporation (ORF) typically deploy a number of IT based strategies for avoidance, prevention or recovery from loss [5]. These archives are engaged in a process of long-term Digital Asset Management (DAM) [11], specifically Media Asset Management (MAM), which focuses on storing, cataloguing and retrieving digital AV content. Several commercial tools exist to support the MAM process, some of which support risk treatment strategies such as keeping multiple copies of each file (redundancy). However, these tools do not include a model of risk. The archive must decide on risk indicators and define the way in which these can be measured in order to monitor them, often using separate tools to do so.

Based on the analysis of threats to digital preservation in the DAVID project [7], [5], [12], it is clear that it is necessary to manage the threats to the workflow processes themselves. In this domain, we can note the following main sources of risk: equipment and tools (hardware, services/systems, software/algorithms), file formats (including implementations of standards and variations between versions), processes and human errors.

Workflows are often used to describe business processes and, increasingly often, are used to automate some or all of the process. Automated workflow execution is possible if the process is specified in a machine-interpretable fashion, such as using BPMN. In Hazard and Operability Studies (HAZOP), risks are seen as inherent in processes, as individual steps may fail, causing consequences for later parts of the process, or if the process is not executed correctly. Risk-aware business process management is critical for systems requiring high integrity, such as archives.

A recent review of business process modelling and risk management research has been conducted by Suriadi et al. [13], identifying three parts to risk-aware business process management:

- Static / design-time risk management: analyse risks and incorporate risk mitigation strategies into a business process model during design time (prior to execution).
- Run-time risk management: monitor the emergence of risks and apply risk mitigation actions during execution of the business process.
- Off-line risk management: identify risks from logs and other post-execution artefacts, such that the business process design can be improved.

Several approaches have been proposed to model business processes and risk information such that it enables risk analysis. Rosemann and zur Muehlen propose integrating process-related risks into business process management by extending Event-driven Process Chains (EPC) [14]. Risks are classified according to a taxonomy including structural, technological and organisational risks.

Analysis of process risks is difficult given that operational risks are highly dependent on the specific (and changing) business context. Many risks are caused by business decisions (e.g., preservation selection strategy or migration path), so large volumes of data required for statistical methods are often not available for analysis. Those who subscribe to this thesis use structural approaches, such as Bayesian networks, HAZOP and influence diagrams. For example, Sienou et al. [15] present a conceptual model of risk in an attempt to unify risk management and business process management using a visual modelling language.

In contrast to the above thesis, some believe that run-time analysis of risks is possible with a suitably instrumented execution process. Conforti et al. [16] propose a distributed sensor-based approach to monitor risk indicators at run time. Sensors are introduced into the business process at design time; historical as well as current process execution data is taken into account when defining the conditions that indicate that a risk is likely to occur. This data can be used for run-time risk management or off-line analysis.

Given that analysis of business processes using structured and/or statistical approaches can reveal vulnerabilities, it is important to control the risk that these vulnerabilities lead to loss. Bai et al. [17] use Petri nets (a transition graph used to represent distributed systems) and BPMN to model business processes and to optimise the deployment of controls, such that the economic consequences of errors (measured as Conditional Value at Risk - CVaR) are minimised.

Using BPMN, the PrestoPRIME project described the preservation workflows that were implemented in the preservation planning tool iModel [18]. It has shown that tools are required to model such generic preservation workflows in such a way that they can be related to specific preservation processes and augmented with information concerning risks.

In the following section we discuss the value of risk management in the domain of digital presentation and the risk management framework proposed in this paper.

IV. Business Process Risk Management Framework

Here, we present a Business Process Risk management framework (BPRisk) developed in the DAVID project (Section IV-C), designed to support the aims and risk management process discussed below in Sections IV-A and IV-B.

A. Aims of Risk Framework for Digital Preservation

Above, we have discussed the motivations for a risk management of business processes, according to the wider challenges in the domain of digital preservation. For digital preservation / archive management, the key actor we are addressing with the proposed risk framework is the preservation expert / specialist, who is responsible for designing workflows for managing and processing digital AV content. We can summarise here some key value-added aims of a risk management framework in the context of digital preservation:

1) Helping preservation experts develop new workflows, especially the early stages of development. Note that the purpose of the framework is not to replace MAM tools (discussed in Section III, above), nor the preservation experts, but to be a value-added tool to assist them.
2) Helping preservation experts optimise workflows (in terms of cost effectiveness and security), considering

also trade-offs where too many corners are cut (to reduce cost), which may lead to increased risk (that, in turn, may lead to a greater cost).

3) Helping preservation experts communicate and justify decisions about choices for elements in workflows. This may be related to arguing expected financial ROI of putting in place certain risk mitigations, for example. By risk mitigation, we here refer to reducing the likelihood or impact of risk.
4) Helping organisations change their processes, as the risk arising from such changes is typically seen as very high, which inhibits change. However, change is necessary to address the issue of format obsolescence.

From an organisational point of view, some of the key reasons to perform risk management can be summarised as follows:

1) Workflows can be large and complex. Therefore, there can be too many variables and options for preservation experts to consider simultaneously to accurately estimate the potential impact of risk.
2) Risk information is typically in experts' heads, which is itself a risk from the organisation's point of view. The risk framework ensures that the knowledge is captured and retained, and is readily available should the organisation be subject to an audit or the expert is unavailable or leaves the organisation.
3) Improve cost-benefit by a) identifying and understanding key vulnerabilities and b) targeting investments to address those vulnerabilities.
4) Move away from "firefighting". That is, organisations may spend more time and resources dealing with issues rather than preventing them in the first place. Risk management is key to prevention, i.e., spending more time in the planning stages to save time and cost on dealing with issues in the future that could be avoided.

It is important to note that the end users of the risk management framework in this context are unlikely to be risk experts. They are domain (preservation) experts, and they will be acutely aware of a wide range of potential issues concerning the preservation workflows they manage. However, the term risk and explicitly managing risk may be entirely unfamiliar and it is important that the risk management framework is suitably designed to aid the domain experts (rather than simply being a risk registry).

B. Risk Management Process

A risk management framework should support a process that promotes best practices to address the aims discussed above in order to reduce the risks to long-term preservation. There is a natural focus on the planning aspects regarding risk management, but we do need to consider the wider context as well.

Several risk standards and methodologies exist, but it is not within the scope here to discuss them in detail. However, we will make reference to one in particular, ISO 31000 [10], to show how it relates to a risk management approach proposed here based on the Deming cycle. The Deming cycle is a four-step iterative method commonly used for control and continuous improvement of processes and products. The four steps are: Plan, Do, Check and Act. For this reason it is also commonly referred to as the PDCA cycle, and is key to, for example, ITIL Continual Service Improvement [19]. In general terms, risk management is a part of continual improvement of processes – preservation workflows in this context.

The ISO 31000 [10] risk management methodology is depicted in Figure 1, below, which depicts the various (cyclic) stages from 'establishing the context' to 'treatment' (of risk). Supporting continual improvement of workflow processes is imperative in digital preservation, as discussed in Section II, as one of the key challenges in this domain is obsolescence and one of the key current risk strategies involving file-replication may not be feasible in the future.

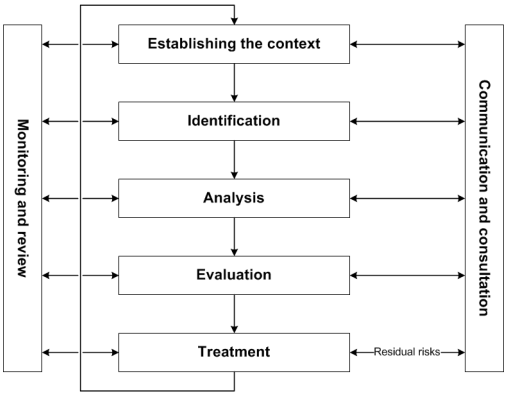


Figure 1. ISO 31000 risk management process.

Given the aims discussed above, each of the four stages of the Deming cycle is covered below from the perspective of what a user (preservation expert) would do, with reference to the related stages of the ISO 31000 methodology).

Plan ('establishing the context' and 'identification' stages of ISO 31000): build workflows, capture risk information, simulate workflow execution scenarios to identify key vulnerabilities and estimate impact of risk, and make decisions.

Do ('analysis' stage of ISO 31000): execute business process, orchestrate services, and record execution meta-data.

Check ('evaluation' stage of ISO 31000): analyse workflow execution meta-data and process analytics, calibrate simulations and trigger live alerts.

Act ('treatment' stage of ISO 31000 as well as feedback and loop-back to the previous stages): adapt workflows and manage risk. Re-run simulations (Plan), enacting the offline changes in the real business process and continues execution (Do) and monitoring (Check).

Note also how this relates to the three risk-aware business processes discussed above from Suriadi et al. [13]; static/design-time risk management (Plan), run-time risk management (Do) and off-line risk management (Check). The final step in the Deming cycle, Act, covers multiple processes.

C. Risk Components

Based on the above aims, a high level component view of the BPRisk framework developed in the DAVID project is depicted in Figure 2. This framework integrates both new components developed in the DAVID project as well as existing open source technologies, which is discussed below.

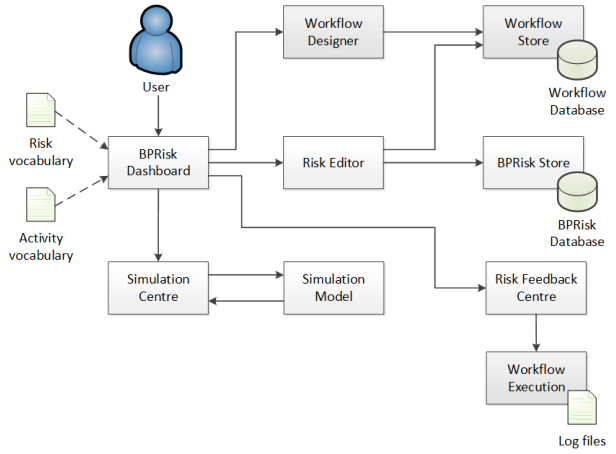


Figure 2. BPRisk framework high level component view.

BPRisk Dashboard: The main entry point for the user from which the user can access the functionalities of the framework, e.g., to create workflows, specify risks, run and view risk simulation results, etc. Figure 2 also shows two vocabularies used, one for known domain-specific risk and one for domain specific activities. This is discussed further below.

Workflow Designer: There are several existing, mature, tools for this, supporting the well-known BPMN 2.0 standard, such as Signavio Decision Manager [20] and the jBPM Designer [21]. The latter has been adopted in the BPRisk framework as it is available as open source.

Workflow Store: This is a component to persist any workflows created, updated or imported. Existing tools, such as jBPM come with multiple persistence options and a RESTful [22] API for accessing and managing the workflows.

Risk Editor: As described above, this component is responsible for allowing users to specify risks. As discussed earlier in this paper, the end-users of this system are not likely to be risk experts. Therefore, the Risk Editor utilises the two vocabularies mentioned above in a semantic risk model, which is used to aid users in specifying risks. See Section V for further discussion.

BPRisk Store: This is a component for persisting risk specifications and risk simulation results (a connection from the Simulation Centre has not been depicted in Figure 2 for the sake of simplifying the diagram).

Simulation Centre: This is a component for managing the running of simulation models for workflows annotated with risk information. This component deals with configuring different simulation scenarios and allows users to visualise and compare the results.

Simulation Model: A stochastic risk simulation model that the Simulation Centre can execute. This component simulates executions of the workflow process and the occurrences of risks defined for the workflow activities. As output, the simulation model gives information on, for example, risk occurrences, time and cost spent on risk, and impact of risk.

Risk Feedback Centre: A component for getting data from real workflow executions that can be used to a) analyse the workflow execution meta-data and b) to modify/adapt/calibrate the workflows (e.g., risk details) and simulation configurations to improve the accuracy for future simulation scenarios.

Workflow Execution: An external software component to the BPRisk framework, which would be invoked to execute a workflow process. This is a source of workflow execution data for the Risk Feedback Centre.

D. Implementation and Integration

A BPRisk framework prototype has been implemented as a RESTful [22] web application using Java Spring [23]. Figure 3 shows an architecture diagram of the key components that are in scope of this paper.

Web services are denoted with [WAR], comprising the BPRisk web application itself, a simulation service, the jBPM Designer (with Guvnor for workflow storage) and a Sesame service for the OWLim triple store used. The BPRisk web application follows a Model-View-Controller (MVC) [24] design pattern, comprising a shared data model (not discussed here), a control layer and view components for User Interface (UI) interactions with the users. Due to the MVC RESTful approach taken, it is possible for external client applications to access the data services, such as workflow data, in way that allows flexible compsition of relevant information for the end-user.

As noted above, the jBPM Designer has been integrated for workflow design, i.e., to graphically create new workflows or editing existing workflows. It supports the BPMN 2.0 standard, which allows users to specify workflows using, e.g., events (such as 'start' and 'end'), activities and connections between the activities, such as sequence flows or gateways to represent process logic. Figure 11 gives an example of a BPMN workflow using exclusive OR gateways.

The jBPM Designer uses the jBPM Guvnor as a workflow store by default, which has been adopted in BPRisk. However, to enable integration with other workflow management tools, the BPRisk framework has been designed such that workflow data is accessed via the Risk Data Service API, as depicted in Figure 3. More details are shown in Figure 4. A generic 'Workflow Accessor' component communicates with a specific workflow accessor module that functions as an adaptor. Here, a 'Guvnor Accessor' is shown, which will make a call to the Guvnor Service via its REST API in order to manage workflow information. Within the Risk Data Service API, the workflow data from Guvnor is processed via an 'Activiti BPMN parser' [25] before workflow information is returned in a shared BPRisk data model format (activities, gates and flows).

There are three different data storages depicted in the Risk Data Service API layer:

1) Workflow Store: for persisting and accessing the BPMN workflows, using the jBPM Guvnor as discussed above.
2) BPRisk Semantic Store: for storing semantic data pertaining to workflows, linking with controlled vocabularies for risks and domain specific activities, enabling semantic reasoning to aid users in creating or optimising their workflows.
3) BPRisk Project Store: for storing all other BPRisk data, including workflow projects, users, simulation configurations and simulation results.

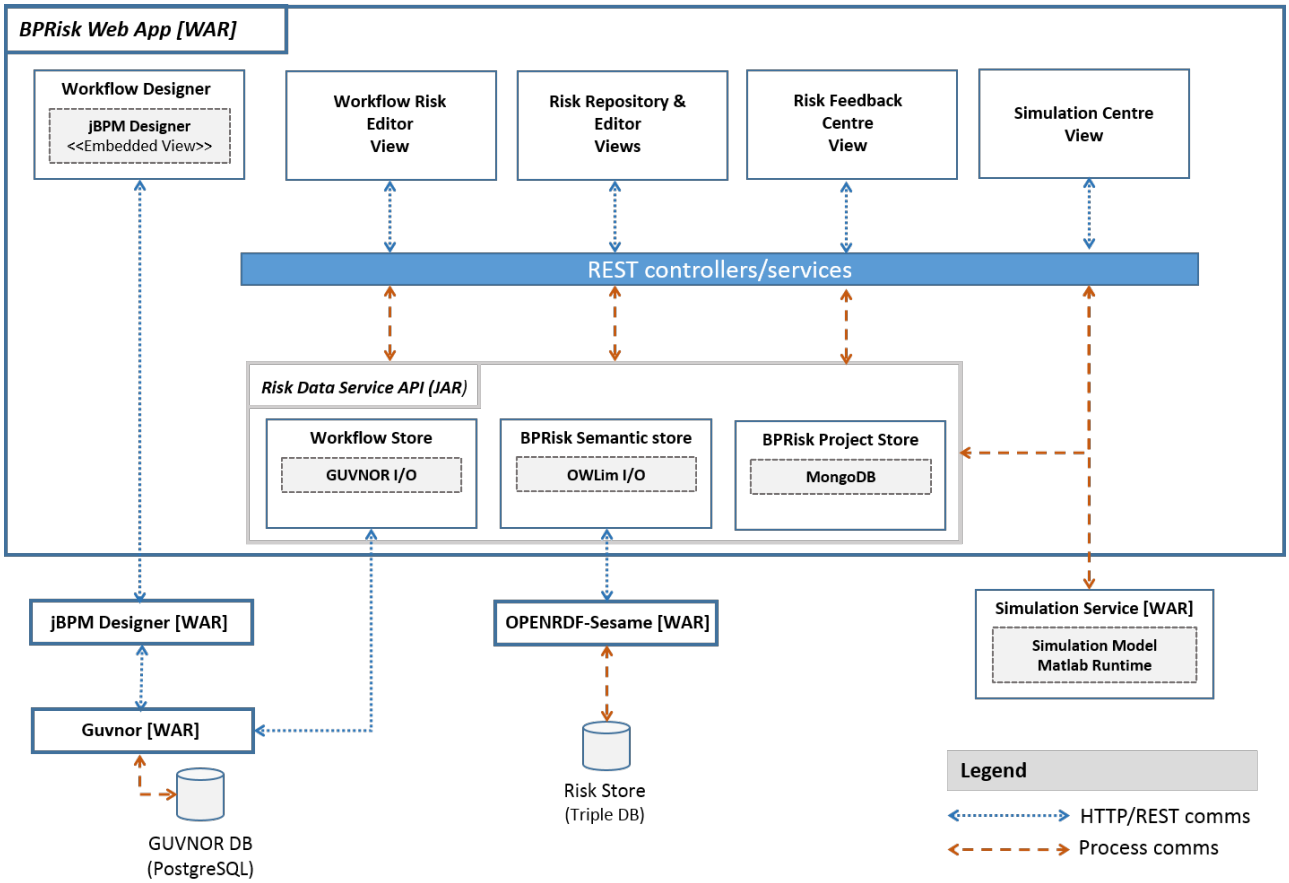


Figure 3. BPRisk architecture diagram.

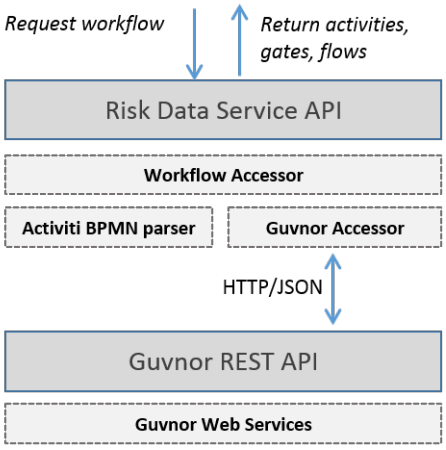


Figure 4. BPRisk workflow API.

A risk simulation model has been implemented in Matlab Simulink [26]. This has been integrated via a separate web service, to enable modularity and scalability as simulations can be computationally heavy. More details of the simulation modelling is provided in Section VI, followed by a discussion of simulation results in Section VII.

V. Semantic Risk Modelling

The BPRisk framework utilises a semantic risk model for specifying and reasoning about risks associated with workflow activities. The modelling approach is generic in nature, utilising a multi-level ontology to include domain specific workflow activities and risks.

A. Modelling Approach

The BPRisk ontology represents information related to risks, controls and activities. This representation allows flexibility and extensibility of the risk model. It can be easily published (e.g., as a set of OWL files), and can be extended in unexpected ways. For example, the BPRisk ontology allows for the possibility of injecting provenance based information that can provide an auditable trail linking the identification of a risk factor (related to a workflow element) to its subsequent treatment using a provenance based ontology such as W3C PROV [27].

The approach to building the ontology is based on work done in the SERSCIS project [28]. The authors use a layered, class-based ontology model to represent knowledge about security threats, assets and controls. Each layer inherits from the layer above. The CORE layer describes the relationships between a central triad (threat, asset, control). A domain security expert creates sub-classes for each of these core concepts to create a GENERIC layer. A system expert further sub-classes the generic concepts to specialise them for the system

of interest, creating the SYSTEM layer. Note that this ontology was used in the context of modelling systems and interactions between system components, where it is assumed that a system of a particular type is always subject to the threats identified by the security and system experts. This expert knowledge, therefore, helps the users create more secure systems as they may not have this expert knowledge themselves.

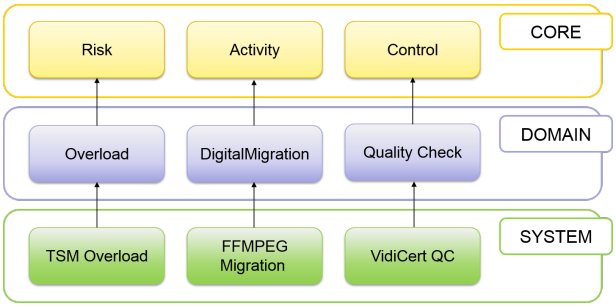


Figure 5. Workflow risk ontology layers.

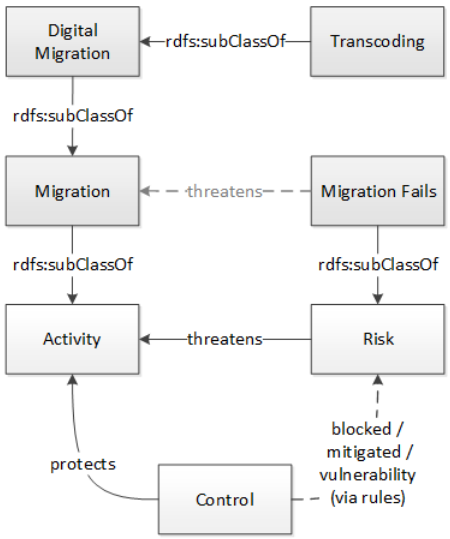


Figure 6. BPRisk ontology with sub-classing examples. CORE layer entities depicted in white and DOMAIN layer entities in grey.

The same, layered, ontological approach has been taken here, as illustrated in Figure 5, though with a few modifications. While the triad in the CORE layer in SERSCIS includes *Asset*, there is only one asset of value in this context – the digital AV object, which can be affected by different *Activities* in a workflow process (e.g., ingest, storage and transcoding). The term *Threat* used in SERSCIS can be understood as *Risk* in this context. Therefore, the CORE layer in BPRisk comprises a triad of *Risk*, *Activity* and *Control*.

The GENERIC layer from SERSCIS has been renamed to the DOMAIN layer here, as it better reflects the level at which knowledge of domain specific (generic) activities and risks are represented. It is at this level, we incorporate controlled domain vocabularies, which are discussed further below in Section V-C. This layer can be further extended via the SYSTEM layer by users of the BPRisk application.

B. Model Definition

The model focuses on the *Activities* in the preservation life cycle and the *Risks* that are inherent in their execution. *Controls* can be put in place to block or mitigate these *Risks*. The CORE layer comprises *Risk*, *Activity* and *Control*, as well as basic relationships such as 'Risk *threatens* Activity' and 'Control *protects* Activity'. However, the relationship between *Control* and *Risk* is established via rules that abstractly encode how types or super-types of both controls and risks can be linked together (see the following section), to determine the appropriate relationship. That is, a *Risk* is only considered *Mitigated* if an appropriate *Control* is in place. This is illustrated below in Figure 6.

The DOMAIN layer has been developed in the DAVID project for digital preservation, which describes common preservation activities, risks and controls. These are modelled as sub-classes, which can be quite hierarchical. As an example, the DOMAIN level classes in Figure 6 include three sub-classed *Activities*, 'Migration', 'Digital Migration' and 'Transcoding', with an associated risk 'Migration Fails'. Migration in this context refers to converting content in one format into another format. Digital migration refers to converting older analogue content into digital form.

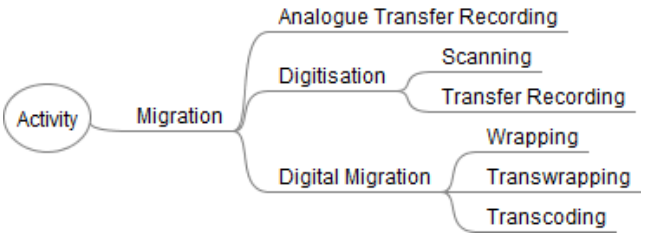
The SYSTEM layer is a further extensible part that would be populated by the users of the BPRisk framework when they build a workflow of specific *Activities* and associate *Risks* to them. For example, a migration workflow may use a specific *transcoding* tool such as FFmpeg [29], which may have specific technical risks not covered by the more generic 'Transcoding' activity. Thus, a 'FFmpeg Transcoding' activity may be added as a sub-class of 'Transcoding' (see Figure 6). This sub-classing is important, as we can reason about risks throughout the hierarchy, as discussed further in Section V-D.

C. Controlled Vocabularies

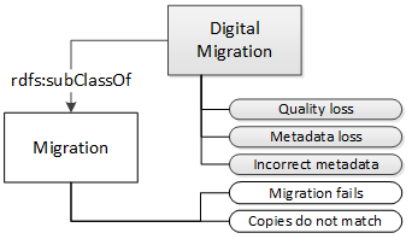
In order to enhance the usability of the BPRisk framework, expert domain knowledge is included via the DOMAIN layer of the ontology presented above in Section V-B. The domain knowledge is incorporated via two controlled vocabularies: 1) known domain activities; 2) known risks and controls to the aforementioned activities.

Although the controlled vocabularies reside within one of the three logical layers in the BPRisk ontology, there can be an extensive hierarchy of entries, which is depicted in Figure 7a. For example, we can see that 'Transcoding' is a type of 'Digital Migration' which is a type of 'Migration' activity. Further, for each activity, the controlled risk vocabulary defines common risks and controls for the known activities, such as for the 'Digital Migration' activity, as depicted in Figure 7b. For further details of the activities, risks and controls in the controlled vocabularies, interested readers are referred to [5].

When a user defines a workflow in the BPRisk framework, the domain knowledge embedded in the semantic model is used as follows. First, the activities from the BPMN workflow are extracted via the Risk Data Service API (see Figure 4). The user then maps each BPMN activity to activities in the BPRisk ontology (using the controlled activity vocabulary). Following this, the user will retrieve suggestions of potential risks, as per the controlled risk vocabulary. For example, for a 'Digital Migration' activity, the users will be presented with five possible risks, two of which are inherited from the parent



(a) Hierarchy of Migration activities.



(b) Risks for the Digital Migration activity.

Figure 7. Examples from controlled vocabularies.

activity 'Migration', as depicted in Figure 7b. The users can then chose which risks apply to the specific activity in the respective workflow.

As noted above, users are also able to add new risks, activities and controls not reflected in the DOMAIN layer. These user-specific additions form part of the SYSTEM layer, extending the knowledge repository. This knowledge is then available to users when working on other workflows. In the following section, we delve into further details on how this functionality is achieved via semantic reasoning.

D. Semantic Reasoning

Providing expert knowledge to users of the BPRisk framework is achieved via semantic reasoning, which is performed according to pre-defined rules.

Rule Composition: The layered abstractions CORE, DOMAIN and SYSTEM in the BPRisk model provide a useful framework within which to characterise generally increasing levels of specialism with respect to workflow activities and their associated risks and controls. Aligned with this arrangement, the rules that create relationships between activities, risks and controls are also expressible within and between these layers. For example, during a video migration activity a risk exists that the copies (that is, the migrated video) will not match the source material. This simple rule is expressed as using the Turtle RDF formalism [30]:

```
:CopiesDoNotMatch a owl:Class ;
  rdfs:subClassOf core:Risk ;
  rdfs:subClassOf [ a owl:Restriction ;
      owl:onProperty core:threatens ;
      owl:someValuesFrom act:Migration
  ] .
```

This risk is applied generally in the DOMAIN level and also to activities derived from the *migration* type - these include *analogue transfer recording*; *digital migration* and *digitisation* activities (as seen in Figure 7a, above). Workflows that include activities that belong to (or are themselves specialities of) this type will automatically generate an instance of this risk when processed by the BPRisk framework. Having identified a risk, one or more controls should be put in place to manage its potential outcomes. Here, the knowledge encapsulated in BPRisk rules can also be used. Activities are said to be 'protected' by controls that are available to manage risk; one very simple protection against a copy mismatch during migration would be to first *check* the migration

```
:CheckMigration a owl:Class ;
  rdfs:subClassOf core:Control ;
  rdfs:subClassOf [ a owl:Restriction ;
      owl:onProperty core:protects ;
      owl:someValuesFrom act:Migration
  ] .
```

and second, *re-do* the migration, if required:

```
:RedoMigration a owl:Class ;
  rdfs:subClassOf core:Control ;
  rdfs:subClassOf [ a owl:Restriction ;
      owl:onProperty core:protects ;
      owl:someValuesFrom act:DigitalMigration
  ] ;
  rdfs:subClassOf [ a owl:Restriction ;
      owl:onProperty core:protects ;
      owl:someValuesFrom  act:Migration
  ] .
```

Both of these control measures would be suggested by the BPRisk framework when the *'copies do not match'* risk is detected. Sometimes more specific knowledge is available that enhances the more general control procedures offered by the framework. In our example, those specific to protecting digital migration activities would be suggested. Below we see that re-introducing missing meta-data is a possible solution for risks threatening digital migration.

```
:ReintroduceMissingMetadata a owl:Class ;
  rdfs:subClassOf core:Control ;
  rdfs:subClassOf [ a owl:Restriction ;
      owl:onProperty core:protects ;
      owl:someValuesFrom act:DigitalMigration
  ] .
```

When a risk has been identified and controls put in place they can be marked up as either *blocked* or *mitigated*.

Rule Encapsulation: Encapsulating the relationships between risks, controls and activities are ultimately encoded as risk classification rules within the ontology knowledge base itself, using SPIN [31]. From a technical point of view, this provides a more flexible method of extending and executing rules incurring zero or only minimal changes to the compiled source used to operate on the results. Running inferencing over the model automatically applies the classification and can also determine the revised state of a workflow when control procedures have been put in place. In our earlier example, we considered the application of controls to act in the presence of risks threatening migration activity. In the SPIN formalism below, we express the fact that the control *'check migration'* blocks the *'copies do not match risk'* that is generated in the presence of a *'migration'* activity.

```
:CopiesDoNotMatch_BlockedBy_CheckMigration_For_Migration
  rdf:type spin:ConstructTemplate ;
  spin:body [
      rdf:type sp:Construct ;
      sp:templates (
          [
            sp:object risk:BlockedRisk ;
            sp:predicate rdf:type ;
            sp:subject [ sp:varName "r"^^xsd:string ; ] ;
```

```
          ] ) ;
        sp:where (
          [
            sp:object act:Migration ;
            sp:predicate rdf:type ;
            sp:subject [ sp:varName "a"^^xsd:string ; ] ;
          ]
          [
            sp:object risk:CopiesDoNotMatch ;
            sp:predicate rdf:type ;
            sp:subject [ sp:varName "r"^^xsd:string ; ] ;
          ]
          [
            sp:object ctrl:CheckMigration ;
            sp:predicate rdf:type ;
            sp:subject [ sp:varName "c"^^xsd:string ; ] ;
          ]
          [
            sp:object [ sp:varName "a"^^xsd:string ; ] ;
            sp:predicate core:threatens ;
            sp:subject [ sp:varName "r"^^xsd:string ; ] ;
          ]
          [
            sp:object [ sp:varName "a"^^xsd:string ; ] ;
            sp:predicate core:protects ;
            sp:subject [ sp:varName "c"^^xsd:string ; ] ;
          ] ) ;
    ] ;
rdfs:subClassOf :RiskClassificationRules ; .
```

In running SPIN rules every time the knowledge about specific workflow activities (contained in the SYSTEM layer) is added we are able to automatically recognise, control and manage risks in a pro-active manner.

As noted above, the SYSTEM layer is developed so that it sub-classes the DOMAIN layer for a specific organisation using the BPRisk framework, as seen above in Figure 6. This should specify the kind of activity in the preservation workflow of interest, e.g., sub-class *Migration* as *DigitalMigration* as seen above in the examples from the DOMAIN layer. Workflow-specific risks can then be automatically generated; for example, the following is a generic construction rule to generate all risks:

```
CONSTRUCT {
    ?uri a owl:Class .
    ?uri rdfs:subClassOf ?gr .
    ?uri rdfs:subClassOf _:b0 .
    _:b0 a owl:Restriction .
    _:b0 owl:onProperty core:threatens .
    _:b0 owl:someValuesFrom ?sa .
} WHERE {
    ?sa (rdfs:subClassOf)+ act:Activity .
    ?sa rdfs:subClassOf ?ga .
    ?gr rdfs:subClassOf core:Risk .
    ?gr rdfs:subClassOf ?restriction1 .
    ?restriction1 owl:onProperty core:threatens .
    ?restriction1 owl:someValuesFrom ?ga .
    FILTER NOT EXISTS {
        ?uri rdfs:subClassOf _:0 .
    } .
    FILTER STRSTARTS(str(?sa),
       "http://david-preservation.eu/bprisk#") .
    BIND (fn:concat(STRAFTER(str(?gr), "#"),
       "_", STRAFTER(str(?sa), "#")) AS ?newclass) .
    BIND (URI(fn:concat(fn:concat(STRBEFORE(str(?sa),
       "#"), "#"), ?newclass)) AS ?uri) .
}
```

This rule finds all activities in the SYSTEM layer and creates a workflow-specific risk for each of the DOMAIN layer risks that threaten the activities' parent class. The name of the workflow-specific risk in this example is generated by concatenation of the DOMAIN layer risk name and the workflow-specific activity name.

E. Discussion

The purpose of the semantic modelling in the BPRisk framework, as mentioned earlier in this paper, is to support the end-users, who are typically not risk experts, in optimising and building more robust workflows in order to ensure the long-term value of their digital content.

Formal representation of domain knowledge (linking activities, risks and controls) using the BPRisk semantic framework confers upon its users the ability to encapsulate and operationalise expertise in the preservation of media in the context of workflow based processes. Knowledge is structured in terms of i) hierarchies that are capable of expressing general and specialisations of cases (activities or risks) and ii) bespoke networks of connected activities, risks and controls that combine to form rules that flexibly express scenarios applicable to a wide range of workflows. In the example provided above, we explore this ability in the scenario where a risk is mapped to the type (and sub-types) of migration. Here, this specific risk is defined as relevant in the context of migration type activities and is managed by a particular recommended control. However, note that the same risk type may also be applicable to other unrelated activities but may not call for the same controls. Managing risk using this formalism, thus, offers the user customisable responses to risk depending on the activity in hand. This has been made possible through the use of an ontological approach to knowledge engineering in which RDF and SPIN technologies have been used to build a knowledge base that is readily extensible by end-users (via the BPRisk user interface). This approach, therefore, adds value to media workflow assets by augmenting them with expertise that can be queried and refined at design time, then tested and updated (through simulation and feedback from real-world exectuion, as described in the following section).

A large part of the knowledge base described here is particular to *media workflow* risk management. However, the application of the BPRisk framework is not limited to this enterprise and could be applied to other problem domains in which risk within workflows play a significant role. The underlying BPRisk architecture and services would remain the same, but it should be noted that a significant initial effort would be required to capture and transform knowledge gathered from experts in order to populate the DOMAIN layer of the ontology with common activities, risks and controls.

VI. Simulation Modelling

In this section, we present the work done on workflow risk simulation modelling, which is an integral part of the BPRisk framework. In respective sections below, we discuss the purpose of the simulation modelling, the risk impact model, the risk generation model and risk control procedures. Thereafter, in Section VII, we will present results from simulation modelling on a workflow at ORF, the Austrian Broadcasting Corporation.

A. Purpose of Simulation

The purpose of risk simulation modelling is to help an organisation to reduce cost by designing or optimising workflows in order to reduce the likelihood or impact of risks occurring. For example, it could be used to help justify expenses on technology and quality control tools, showing the anticipated cost of dealing with issues (risks) when they are not addressed

(controlled) versus the cost of preventing them. The costs may be less, so we can say there is a ROI. The simulations can help identify the key vulnerabilities in a workflow and to help target investments.

The aim is to expose issues at design-time before a workflow is actually executed. In this paper, the risk simulation addresses issues that could occur in activities conducted in preservation workflows. There could be technical risks, such as a system operation failing, or human errors such as mistakes being done because the person is overloaded by too much content to deal with.

To this end, a stochastic risk management model was developed. This model allows users to simulate different scenarios and to produce confidence intervals for different risk measures, if required, by means of Monte Carlo simulations. Moreover, the stochastic model allows end-users to explore 'what if' scenarios and can be used both during planning and operation stages. The proposed stochastic risk management model consists of three main parts:

- Risk Impact Model.
- Risk Generation Model.
- Risk Control Procedures.

Below we describe each of these parts in detail.

B. Risk Impact Model

To classify possible risks (threats) in digital preservation, we have adopted the Simple Property-Oriented Threat model (SPOT) for Risk Assessment. The SPOT model [32] defines six essential properties of successful digital preservation: Availability, Identity, Persistence, Renderability, Understandability, and Authenticity. Interested readers are referred to the original article for details. However, we will give a short definitions of each property and list threats associated with this property.

Availability is the property that a digital object is available for long-term use. *Threats*:

- A digital object deteriorated beyond restoration power.
- Only part of the digital object is available for preservation.
- A digital objects is not available for preservation due to disappearing, cannot be located or withheld.

Identity is the property of being referenceable. A limited amount of metadata is required for this property. *Threats*:

- Sufficient metadata is not captured or maintained.
- Linkages between the object and its metadata are not captured or maintained.
- Metadata is not available to users.

Persistence is the property that the bit sequences continue to exist in usable/processable state and are retrievable/processable from the stored media. *Threats*:

- Improper/negligent handling or storage.
- Useful life of storage medium is exceeded.
- Equipment necessary to read medium is unavailable.
- Malicious or/and Inadvertent damage to medium and/or bit sequence.

Renderability is the property that a digital object is able to be used in a way that retains the object's significant characteristics (content, context, appearance, and behaviour). *Threats*:

- An appropriate combination of hardware and software is not available, cannot be operated or maintained.
- The appropriate rendering environment is unknown.
- Verification that a rendering of an object retains significant characteristics of the original cannot be done (e.g., a repository is unable to perform sufficient quality assurance on migration due to volume).
- Object characteristics important to stakeholders are incorrectly identified and therefore not preserved.

Understandability requires associating enough supplementary information with archived digital content such that the content can be appropriately interpreted and understood by its intended users. *Threats*:

- The interest of one or more groups of intended users are not considered.
- Sufficient supplementary information for all groups of intended users is not obtained or archived.
- The entire representation network is not obtained or archived.
- Representation network of supplementary information is damaged or otherwise not renderable in whole or in part.

Authenticity is the property that that a digital object, either as a bitstream or in its rendered form, is what it purports to be. *Threats*:

- Metadata and/or documentation are not captured.
- Metadata maliciously or erroneously describes the object as something it is not.
- A digital object is altered during the period of archival retention (legitimately, maliciously or erroneously), and this change goes unrecorded.

Since not all possible threats/risks in digital preservation workflows may fall in the six properties mentioned above, we introduce an extra possible state in the SPOT model for such cases: ***Other***.

C. Risk Generation Model

The stochastic risk generation model is based on simulating a workflow in which risks associated with workflow activities take place based on risk occurrence probabilities and dependencies between risks. Dependencies between risks can be within a single activity or between consecutive activities. Below, we given an overview of the data that is needed for workflow simulation, divided into the following categories for convenience: general, workflow-related, risk-related, control-related and other simulation parameters.

General data:

- The purpose of the workflow under consideration. That is, what the workflow does, inputs and outputs to/from the workflow.
- The objectives of risk analysis for this workflow.

Workflow-related data:

- List of activities in the workflow, a short description of each activity, and how the activities are connectedto each other.
- Decision points in the workflow, and, based on records or previous experience, how often each decision are usually made at each decision point (e.g., at decision point 1, D1 will be made approximately 90% of the time and D2 10% of the time).

Risk-related data (for each activity):

- List of risks (threats) that can take place and their descriptions.
- Any dependencies between the risks in the same activity and/or risks from different activities. E.g., can the risks in the same activity occur simultaneously?
- Frequency of each risk occurrence, either from records or estimated based on the previous experience; frequencies of more than one risk taken place in an activity if relevant; any changes in frequency of occurrence of some risk in the activity if other risk in the same activity took place.
- For each risk
  - Probability (frequency) of occurrence.
  - Detection level (if known).
  - Negative impact on workflow measured in monetary values, percentages or some impact scale.
  - Affected SPOT properties.
- If combination of risks can occur:
  - Frequency of combined occurrence.
  - Multiplication factor, which is used to update the probability of a risk if combinations of risks occur either in the same or previous activity.

Control-related data (for each risk):

- Is anything done on the fly (Ad-Hoc Control)? If yes, is the Ad-Hoc Control procedure covered by budget overheads? How effective is the Ad-Hoc Control procedure (a value for 'Expected Success' would be provided)?
- Are Active Control procedures available for a given risk and activity? If so, is there a delay before the Active Control takes place?
- List all other control procedures dealing with this risk and their effectiveness.
- If more than one procedure dealing with risk is available, describe conditions when different procedures are activated.
- List costs associated dealing with risk and time spent on dealing with risk.
- Describe how negative impact is reduced when Ad-Hoc or/and other control procedures are applied.

Other simulation parameters:

- Number of items to be processed through a workflow.
- Annual throughput of items.
- Number of items to be processed during a day, week, month or year.

The risk occurrence probability is calculated based on the Estimated Frequencies (EFs) of risks provided by a user. We assume here, that EFs of risks are based on a pre-defined annual throughput of a given workflow, and, therefore, a risk occurrence is simulated on a per item (digital AV file) basis. If a pre-defined annual throughput of the workflow changes, the new estimated frequencies can be updated using Estimated Frequency Factor (EFF) provided by the user. For example, if we are interested in processing N items per month, which is equivalent to $12 \times N$ items per year, then the EF for risk i in activity j can be calculated as:

$$EF_{new}(i,j) = 12 \times N \times EFF(i,j) \quad (1)$$

where $EFF(i,j)$ is the Estimated Frequency Factor for risk i in activity j.

$EF_{new}(i,j)$ represents a frequency per year in this form.

The probability of risk occurrence based on EF per item can be calculated as follows.

$$P(risk\,for\,1\,item) = \frac{EF}{N_{items}} \quad (2)$$

where EF is the estimated frequency based on a pre-defined throughput for a given workflow in pD (per day), pW (per week), pM (per month) and pY (per year).

N_{items} corresponds to a number of items that can be processed in a day/week/month/year, based on this pre-defined throughput.

If the throughput of items is changed, then instead of EF, EF_{new} *will be used in conjunction with new values* pY, pM, pW *and* pD.

By the nature of dependency between risk occurrences, all risks can be divided in the following groups:

1) Risks do not have any known dependency between each other and it is assumed that they cannot occur simultaneously.
2) The frequency of one risk in a given activity changes temporally if another risk in this activity took place. In this case these risks can occur simultaneously. Otherwise, both of the risks can occur only one by one (mutually exclusive).
3) The frequency of ***Risk A*** in the next activity changes temporally if ***Risk B*** in the previous activity occurs. This situation will be modelled as follows: if ***Risk B*** took place, then the frequency of ***Risk A*** is changed accordingly (for a single run of the workflow).
4) The risks can occur simultaneously. In this case the frequencies of each risk and the frequency of risks occurring simultaneously are used to simulate such a situation. In this case 'Estimated Frequency' means that only a given risk took place, 'Frequency of Combined' shows the joint frequency of risks.

The identification of risk generation groups will be done automatically by checking corresponding values in order of priority:

1) Multiple Risk-Entry per Activity.

2) Multiplication Factor.
3) Frequency of Combined.

A Detection Level parameter allows us to simulate whether a risk was detected or not. If a risk is detected, then the procedure for dealing with risk will be put in action (see the next section). Otherwise we mark affected SPOT properties and record level of Negative Consequences (NC). An example of risk specification and related simulation configuration is given in Table I.

D. Risk Control Procedures

The stochastic risk management model implements a procedure of dealing with risk that comprises two types of controls: **Ad-Hoc Control** and **Active Control**. These controls only apply if a risk is actually detected. The schematic presentation of Risk Control procedures is shown in Figure 8. If a risk is detected, then the Ad-Hoc Control procedure is started. Active Control is applied only if a cost spent on an Ad-Hoc Control procedure is higher than a pre-defined value. This is generally only likely to be issued for large and significant risks and could be, for example, re-training staff or allocating more resources. This type of control would typically incur additional cost and time to be put into place. However, note that Active Control is not necessarily available for all activities/risks. In this case only the Ad-Hoc Control procedure is applied to dealing with those risks.

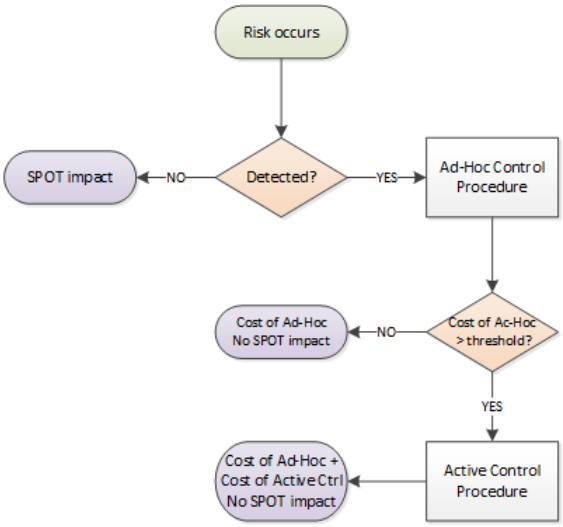


Figure 8. High level risk control flow chart. For simplicity, negative consequences are not shown, but do occur along with 'SPOT impact'. Similarly, time spent on dealing with risk is omitted; just referring to financial cost.

The details of the Ad-Hoc Control procedure are illustrated in Figure 9. It can be covered by overhead or not. Overhead is a term used here for either a budget or a percentage of resources set aside *a priori* to cover the cost of dealing with issues. If not covered by overheads, it will result in cost and time spent with dealing with the risk. However, if the Ad-Hoc Control procedure is covered by overhead and has 100% Expected Success of Ad-Hoc counter-measures, then a) the risk does not have any effects on the assets properties (SPOT model) or Negative Consequences and b) there are no (extra) costs associated with dealing with the risk.

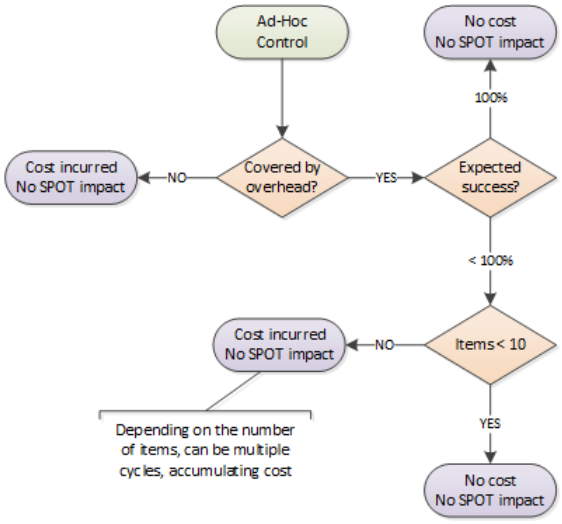


Figure 9. Ad-Hoc Control procedure flow chart. For simplicity, negative consequences are not shown, but do occur along with 'SPOT impact'. Similarly, time spent on dealing with risk is omitted; just referring to financial cost.

If the Expected Success of the Ad-Hoc Control procedure is not 100%, then, based on the number of items to be processed through the workflow and Excepted Success rate, additional Ad-Hoc Control procedures are performed as follows.

1) For up to 10 items: the 1st attempt of the Ad-Hoc Control procedure always successful independent of the Expected Success rate provided for the respective risk.
2) For 11 to 100 items: 2nd attempt will be always successful if the Expected Success rate is 50% or above. 3 attempts will have to be made otherwise to achieve success.
3) For 101 to 1,000 items: 3rd attempt will be needed to reduce the Negative Consequences to zero.

Note that, in this case, the Ad-Hoc Control procedure is performed until Negative Consequences is zero or near zero. In general, the number of attempts needed is equal to the order of the number of items to be processed via the workflow. Let us take an example:

A risk is detected for 500 items, and we have a 90% Expected Success rate for this risk.

This means the 1st attempt will resolve the issue for 450 items → 50 items remaining.

The remaining 50 items are subject to a 2nd attempt → 5 items remaining.

The remaining 5 items are then subject to a 3rd and final attempt, and for up to 10 items, we have modelled the attempt to have a 100% success rate, regardless of the value provided for the Expected Success rate.

For each attempt, time and cost is accumulated, and it is this sum that is subject to the Active Control check; i.e., if this exceeds some pre-defined threshold.

The general formula for calculating costs of the Ad-Hoc Control procedure is as follows:

Table I. Risk specification example.

	TSM Retrieve		Mapping Control	
	Wrong file selected	***Retrieve fails***	***Overload***	***Wrong assessment***
Estimated Frequency	5 per year	5 per year	2 per month	2 per month
Estimated Frequency Factor	0.0004166	0.0004166	0.002	0.001
Multiplication Factor	1	1	1	5
Multiple Risk Entry per Activity	No	No	Yes	Yes
Frequency of combined	None	None	None	None
Detection Level	90%	100%	100%	75%
Level of Severity	1	1	2	2
Expected Success of Ad-Hoc counter-measures	100%	90%	50%	90%
Cost associated with Risk (CAR per hour)	€50	€50	€70	€50
Time spent on dealing with risk (TAR per item)	0.1 hrs	0.2 hrs	0.5 hrs	0.2 hrs
Cost for Active Control strategy (CACS)	€800	None	€1,000	€800
Active Control Activation rule	$(CAR \times TAR) \times 1.2 > CACS$		$(CAR \times TAR) \times 1.0 > CACS$	$(CAR \times TAR) \times 1.2 > CACS$
Delay of Active Control (days)	5	0	5	22
SPOT Availability	1	1	1	1
SPOT Identity	0	0	0	0
SPOT Persistence	0	0	0	0
SPOT Renderability	0	0	0	0
SPOT Understandability	0	0	0	0
SPOT Authenticity	1	0	1	1

$$cost = TAR \times CAR \times \sum_{i=1}^{k} n_i \quad (3)$$

where TAR is the time needed for dealing with risk for one affected item,

CAR is a cost associated with dealing with risk for 1 hour TAR.

k is a number of attempts of the Ad-Hoc Control procedure calculated as described above, based on a number of items passing through the workflow.

n_i is a number of affected items after each Ad-Hoc effort calculated as a product of the number of affected items before the i^{th} attempt and (1 - success rate of Ad-Hoc) in decimal points.

Active Control is applied only if a cost spent on Ad-Hoc Control exceeds a pre-defined allocation of available resources for Ad-Hoc Control (CACS). Active Control does not need to be defined for all activities/risks. However, if Active Control is available then a check is needed for its activation. The activation of Active Control is possible after any number of Ad-Hoc Control procedures according to the following formula:

$$[TAR \times CAR \times (n_i + n_r)] \times k > CACS \quad (4)$$

where k is an activation coefficient.

n_i is the number of affected items in the i^{th} Ad-Hoc attempt.

n_r is the number of remaining items that have to be processed during delay of Active Control.

An additional check has to be performed if $n_r < \frac{PID}{2}$, then Active Control is suspended (called off) even if the condition in Equation (4) holds. PID is a number of items which can be processed during the delay of conducting Active Control. For example, if 'Delay of Active Control' is 1 week, PID = 230. Then, if after 1 Ad-Hoc Control procedure, 100 affected items are left, no Active Control is activated.

If the condition in Equation (4) is true, then Active Control will be applied. Otherwise the Ad-Hoc Control procedure is used. In case of applying the Ad-Hoc Control procedure, NC is zero and no SPOT properties are affected. Since Active Control incurs a delay, the cost of dealing with risk is calculated as follows:

$$cost_{risk} = cost_{ad-hoc} + cost_{ad-hoc-pid} + CACS \quad (5)$$

where $cost_{ad-hoc}$ is the cost of the Ad-Hoc Control procedure before the activation of Active Control.

$cost_{ad-hoc-pid}$ is the cost of the Ad-Hoc Control procedure during delay before Active Control has effect.

Active control will be activated for a given activity only if a sufficiently large number of items will pass through an activity. For the example workflow scenario discussed in the following section, a threshold of 100 files was chosen according to the practices at ORF.

VII. BPRisk Application and Results

In this section, we give an example of how the BPRisk framework has been applied in the design of a workflow in collaboration with the Austrian Broadcasting Corporation, ORF. Respective sections below present a workflow, simulation scenarios, results from simulation modelling and an discussion by ORF to evaluate the accuracy and value of the simulation results.

A. MXF Repair Workflow

Within the DAVID project, the BPRisk framework has been developed with use cases from both the French National Archive (INA) and the Austrian Broadcasting Corporation (ORF), such as planning for migration of old, analogue, content into new, digital, formats (digital migration). Here, we include an example of the use of BPRisk in the planning of an MXF Repair workflow at ORF, which has been used within the DAVID project for validation purposes.

MXF is an abbreviation for a file format; Material eXchange Format. The standard for its use is ambiguous in places and some tool implementations are inconsistent. The result is format compatibility issues, i.e., the files may not standard compliant and, therefore, may not be possible to play in the future. The MXF Repair workflow uses a service called CubeWorkflow [33], which analyses media files for compatibility issues at the file wrapper and bit stream levels. For this scenario MPEG-2 [34] encoded (bit stream) MXF (wrapper) D-10 (SMPTE 356M) [35] files were used. A logical view of the different layers of a digital AV file is illustrated in Figure 10. After the workflow design (planning) was completed, the workflow was executed and the results of the planning could be compared with the monitoring data collected during its execution (see Section VII-D, below).

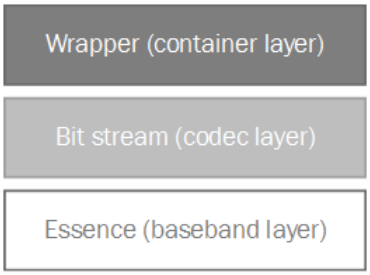


Figure 10. Logical view of layered structure of AV files.

The MXF Repair workflow is depicted in Figure 11, which consists of 9 activities and 2 exclusive OR gateways (both pertaining to points where errors may be detected and handled) Each activity is briefly described as follows:

TSM Retrieve: an activity representing the retrieval (download) of MXF files from a Tivoli Storage Management system (TSM; an LTO tape based IT storage unit).

CubeTec Repair Server Input-Share: network storage on the CubeTec Repair Server to store the retrieved MXF files from the previous activity in order to be processed by the Cube Workflow system.

Cube Workflow: this activity represents a black-box of the Cube Workflow system executing a general analysis (of the Wrapper and Streams) of the MXF files in the INPUT-Share (previous activity). This analysis will detect relevant file errors, attempt to repair errors, and conduct a final control/analysis check of the repaired files. The MXF files passing the final check will be transferred to the ESYS Input-Share.

ESYS Input-Share: network storage in the ESYS Serverframework to gather files for Upload. ESYS = Essence Storage System (by IBM), used by ORF.

Upload: general storage step of ESYS, where MXF files are written to LTO tapes in ESYS and registered in FESAD (the TV Archives MAM), including the production and registration of preview files and keyframe light tables.

Mapping Control: this activity is a manual quality control of the automatic mapping results from the upload process, which is conducted by a person. This is done by comparing the file content (video, audio) with the descriptive metadata in the FESAD entry.

Repair: this is an optional step. If the previous Mapping Control revealed a mapping error, then a manual "reallocation" of the affected file(s) from the wrong FESAD entry to the correct one is performed.

Preview Alignment: manual setting of correct IN and OUT markers of video footage via a special Preview Alignment Tool to mark the beginning and end of specific sections. This step is needed to avoid wrong preview ranges in clustered contents and alike.

Repair / Adjustments: this final activity covers other necessary manual repairs in descriptive and technical metadata (this is a general activity at ORF whenever major changes are done in a FESAD entry).

This is a small workflow, which is ideal for validation and visualisation to help clarify aspects of the risk specification and role of workflow simulation in the BPRisk framework. However, workflows can be significantly larger and more complex, which is a trend we can expect to continue in the future given the adoption of automated tools for carrying out workflow activities in the media industry. This, in turn, will increase the demand for tools to help with workflow planning and analysis.

In the DAVID project, the DOMAIN layer of the BPRisk ontology has been created based on controlled vocabularies for preservation activities, tools and risks, which has been discussed above in Section V-C. Each of the workflow activities have been mapped to activities in this controlled vocabulary. For example, the first activity in the workflow, 'TSM Retrieve', maps to 'Acquisition/Recording' in the preservation vocabulary. And two risks have been identified for this activity: a) wrong file selection and b) retrieve fails. The semantic reasoning rules discussed above, in Section V, enables the BPRisk framework to prompt users with such risks at design time when they add the activity to the workflow they are designing.

After specifying risks for the different activities, workflow simulation scenarios were set up with ORF for this workflow. To simulate workflow execution, additional parameterisation is required, such as estimates for how often the risks are likely to occur, and the expected time and costs for dealing with any issues that may occur. Values were set based on the experiences the workflow and technical experts at ORF had of the tools and activities used in the workflow, as well as observations from monitoring data where available. In the future, these estimates are intended to be updated and improved via the Risk Feedback Centre, as discussed above in Section IV-C.

B. Risk Simulation Configurations

In the MXF Repair workflow, each activity has got two risks that can occur. The risks are a mixture of system and human errors. For example, the 'Retrieve Fails' risk for the 'TSM Retrieve' activity is caused by technical error, while the

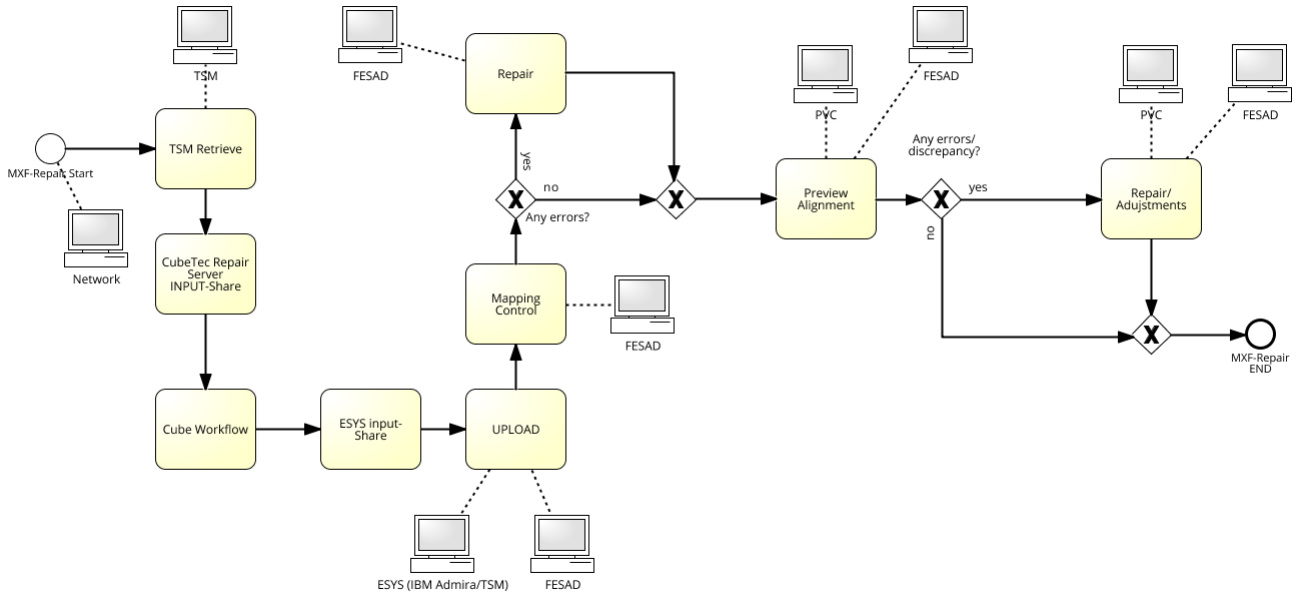


Figure 11. MXF Repair workflow.

'Wrong Assessment' risk for the 'Mapping Control' activity is caused by human error.

For some activities, the two risks are mutually exclusive, i.e., they cannot occur simultaneously; e.g., for the 'TSM Retrieve', 'Cube-Tec Repair Server INPUT-Share' and 'ESYS input-Share' activities. The activity 'Upload' has got two risks that can either happen independently or simultaneously, which requires a combined frequency risks occurrence to be provided for this activity. Moreover, if the risk 'Copy Error' takes place in the previous activity ('ESYS Input-Share'), then the frequency of the risk 'Fails' (for the 'Upload' activity) increases slightly (temporally). For the rest of the activities, risks have the following dependencies: initially the risks are modelled as mutually exclusive, however, if the 'Overload' risk takes place then the frequency of the second risk increases by the given Multiplication Factor and occurrence of this second risk is simulated with the new frequency. That is, if 'Overload' occurred, it can cause the second risk such as wrong assessment or wrong mapping to take place too.

After the activity 'Mapping Control', the simulation procedure will check whether any errors took place (the first XOR gate). If there are errors, then the 'Repair' activity is performed. The probability of any errors is 0.1%, i.e., the 'Repair' activity will take place very seldom during the workflow runs. A condition after the 'Preview Alignment' activity checks whether any error or discrepancy is present. It is known that probability of this error/discrepancy is 60%.

The Estimated Frequencies for each risk are estimated based on an annual throughput of approximately 12,000 files (1,000 monthly, 230 weekly, 46 daily). It is assumed that the MXF Repair workflow runs 15 working hour per day, 5 days a week, 22 working days in a month and 264 working days in a year.

C. Simulation Results

Two simulation scenarios were explored for this workflow, since ORF were interested in a comparison between what is currently done to control risk and a worst-case situation in which no risks were controlled. The 'no control' scenario in this sense serves to demonstrate the importance of what is currently done. For this purpose we ran simulations with 1,000 items (files) processed via a workflow 10,000 times for statistically robust outputs. Given the throughput figures presented above, this equates to 22 days of executing the workflow. The results discussed here are focused on demonstrating the type of insights the simulation modelling gives users in terms of performing risk management for business process workflows. Due to commercial sensitivity, certain parameters such as the financial cost of dealing with risk, are not genuine.

When a simulation is executed in the BPRisk framework, the UI has been designed to give an overview of key facts. For example, for a simulation scenario of the existing set-up for controlling risk:

Risk affecting the most items (**88**): **Upload Fails**.

Total cost of risk treatments: **€1,424.93**.

The most expensive risk (**€965.16**): **Upload Fails**.

Total time spent on dealing with risk: **26.13 hours**.

The most time consuming risk: **Upload Fails**.

Main issue caused is loss of **Authenticity**.

In terms of identifying key vulnerabilities and determining where to target investments, the above summary gives a strong indication towards the 'Upload' activity. From that point, users can explore the data in more detail. Figure 12 shows how many times risks occurred for each activity and how many risks were undetected overall (mean values for the 10,000 Monte Carlo runs). Although the 'Fail' risk for the 'Upload' activity was flagged in the summary as the most

expensive risk, both in terms of time and cost, this figure shows that 'Preview Alignment' had the most risk occurrences (followed by 'Upload' and 'Mapping Control'). We will return to the reason why 'Preview Alignment' was indeed not the most expensive risk below. 'TSM Retrieve' and 'Cube-Tec Repair Server Input-Share' have the smallest number of risk occurrences. No risks took place for the 'Repair' activity, since it is a rare event and it was not evoked even a single time during the simulation. We also observe that a certain proportion of risks are be undetected. This will incur SPOT impacts, which we will return to below.

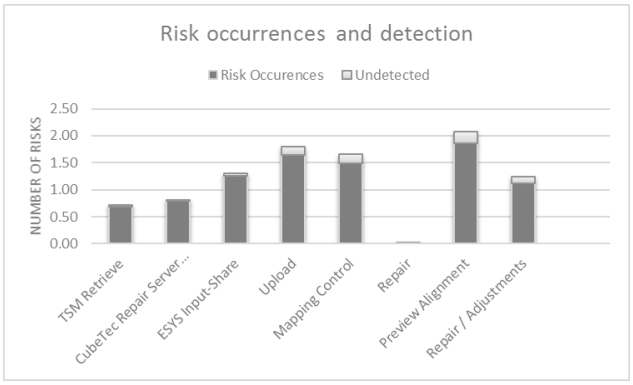


Figure 12. Risk occurrences and detection.

Although the 'Preview Alignment' activity incurs the most risk instances, the risks occurring during the 'Upload' activity will affect the largest number of items; 88 files on average (out of the 1,000). This is significantly more than any other activities, which is depicted in Figure 13. In comparison, the number of affected items during 'Preview Alignment' was 6 on average, and 3 for 'Mapping Control'.

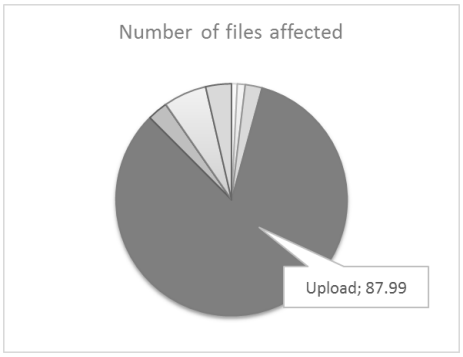


Figure 13. Number of files affected by risk.

Next, we can investigate how the Ad-Hoc Control procedure and Active Control cope with these risks. Figure 14 shows a sum of Negative Consequences (NCs) for each activity with and without Ad-Hoc Control. NC is calculated based on the 'Level of Severity' parameter (defined in the range {1,3}), by summing up this pre-defined value for each risk occurrence (for each activity) that was undetected. For example, the LS for the Upload Fails risk is 1 and the risk occurred 1 times on average, giving a NC value of 1 without control. Mapping Control risks have defined LS as 2, and the 'Wrong Assessment' risk occurs 0.63 times on average, giving a NC value of 1.26 without control.

Not surprisingly, the difference in NC between the two scenarios is significant. If no Ad-Hoc Control procedure is in a place, then the largest NC from risks will be for 'Mapping Control', 'Preview Alignment' and 'ESYS Input-Share'. The Ad-Hoc Control procedure reduces NC to zero for all risks that have been detected. However, for risks such as 'Wrong Assessment' for the 'Mapping Control' activity, the Detection Level is set to 75%, giving a small NC value of 0.32 as the risk only occurred 0.63 times on average during the simulation. The top three activities in terms of the highest NC is almost identical when Ad-Hoc Control is applied; the difference is that the 'Upload' activity has replaced 'ESYS Input-Share'.

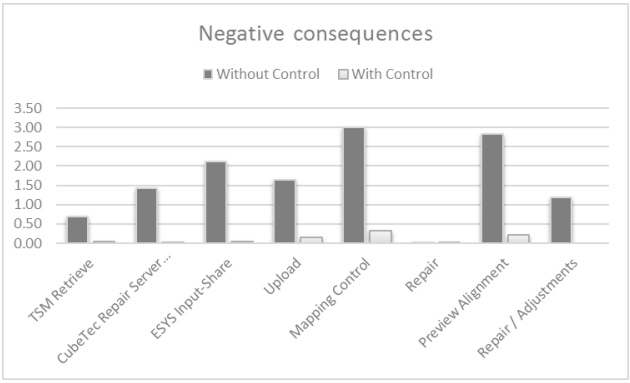


Figure 14. Negative consequences with and without Ad-Hoc Control.

Undetected and uncontrolled risks will have an impact, which we represent according to the SPOT model (discussed above in Section VI-B). For this simulation, the SPOT impact can be seen in Figure 15, comparing the two scenarios with– and without control. The values in this figure are calculated similarly to NC, above, by summarising the number of times risks have been either undetected or have not been successfully addressed in a control procedure. This gives an appreciation of the types of issues to the digital content, caused by the various risks. For example, when risks are controlled, the main issue is loss of authenticity, which is interesting for this type of workflow. Authenticity, as described in Section VI-B refers to the digital content indeed being what it purports to be. Considering that this workflow addresses format compatibility issues, which includes issues such as incorrect or inconsistent meta-data to describe digital content, authenticity being the main issue despite controlling the known risks is interesting as it reflects that the repair processes itself is not 100%.

The Ad-Hoc Control procedure is successful for this MXF Repair workflow in dealing with risks. As seen in Figure 15, the SPOT impacts are minor when the risks are controlled. However, controlling risks incurs costs, both in terms of time to deal with the risk occurrence, as well as financial cost. Figure 16 shows a pie chart that illustrates well the proportional differences in financial cost for addressing risks for the different activities, in which the Upload activity accounts for approximately 68% of the total costs. Moreover, addressing the risks to this activity takes 19.30 hours on average, which is approximately 2.5 working days out of the 22 days of simulating this workflow execution.

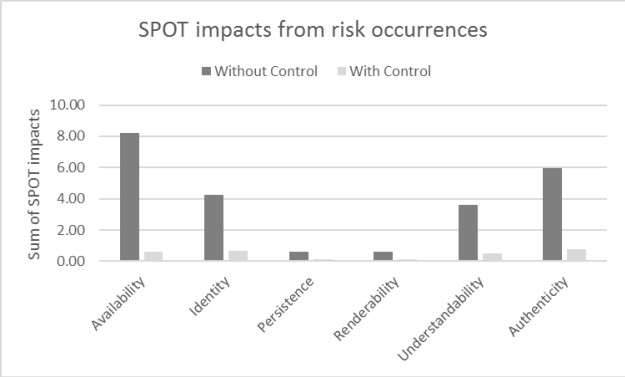


Figure 15. SPOT properties affected by risk (impact - with and without control).

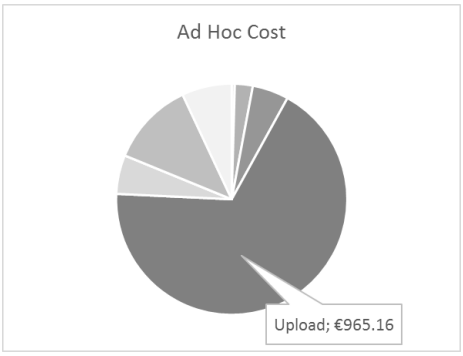


Figure 16. Cost of Ad-Hoc Control.

Active Control was only activated once, during the 'Repair Adjustment' activity. This was for the risk 'Overload', which resulted in a cost of €500 in this simulation scenario. Although the Ad-Hoc Control procedure of dealing with risks is very effective in this particular workflow, it is of interest to rank risks according to impact on a workflow based on a generic score. Table II shows the ranking of risks according to impact score, which was calculated as the product of the number affected items and negative consequences, based on the mean values from the 10,000 simulation runs of the ORF MXF Repair workflow under condition that no Ad-Hoc Control procedure was in place. The ranking is an approach to presenting the risk simulation results in a way to help identify prioritisations of which risks to address. Not only does this representation identify which activities are the most vulnerable to risk, but also we can see that the top 5 highest ranked risks are of the same type - overload. This kind of information, therefore, indicates there is a common technological issue with the systems used not being able to handle the workflow. There may, instead, be alternative systems available, which could be considered in different workflow scenarios to then compare potential ROI, e.g., in terms of whether there is financial gain in upgrading a system if it reduces the risk occurrences (and, thus, the time and costs associated with addressing the risk occurrences).

D. Evaluation and Discussion of Simulation Results

Based on the results discussed above, this section is an evaluation technical experts (workflow designers) from ORF based on the observations they have made in reality. Both ranking of risk/activities and the impact is giving an exact picture of the actual situation ORF experienced when running the MXF Repair workflow. For example, they experienced several upload-fails and 'Overload' was indeed the greatest issue in 'Preview-Alignment'.

Even the ranking from the simulation results (Table II) is nearly identical to ORF's experience. The top three are identical. Thereafter, there are just a few instances of risks that have a rank difference of 1 place. Even at the very end of the ranking the results match ORF's experience, which impressed their workflow-designers very much. Also, in this section of the results, there is only one swap in ranking positions. In the actual workflow 'Retrieve Fails' was slightly higher than 'Wrong File selected'. This may be due to the fact that in the early part of the MXF Repair process they experienced some unanticipated network-issues.

The results for risk occurrence and the number of affected items show a similar picture; the results of the simulation match with our experience during the actual workflow execution. Just the absolute number of affected items for 'Upload' 'Fails' seems to be too high, which is likely due to several changes in the affected ESYS-systems in the period from which values for the simulation model were estimated.

The effect of Ad-Hoc Control, shown in Figure 14, is a very strong argument and help for implementing a proper instalment for control measurements. And again the figures for NC in the different activities reflect very well the actual situation in the MXF Repair workflow; very interesting is the rather high effect of "technical" measurements (e.g., in ESYS Input Share) compared to those with "human-related" measurements (e.g., Mapping Control).

The high impact of Ad-Hoc Control reflects again the actual experience and strengthens our arguments for "investing" in those by having a significant overhead budget available. Especially the results in the SPOT model for Ad-Hoc Control are very impressive and promising. Finally, the results for costs are significant assets for putting forth arguments in budget discussions for future workflows in the domain. The extremely positive impact of Ad-Hoc Control was always neglected or doubted by the decision-makers at ORF.

It has to be stressed here, that the quality of the results are based not only on the model, but also on the quality of the input-data given. ORF put significant effort and time in providing accurate and reliable data for both the model and the simulation, so it has to be expected that the old archive-rule "Garbage in = Garbage out" is valid for risk management and assessment as well. You have to invest a good percentage of the budget reserved for "accompanying measures" for this activity, which included the calculation and surveying of quality planning data to get good results and actually have the chance to save money in the actual workflow or process. And being an expert or involving experts in the domain is highly necessary to save efforts in this process of planning and data collection.

For all colleagues at ORF involved in the process, the tool proves to be a very good and highly reliable instrument to

Table II. Risk ranking according to simulation results and ORF - in descending order.

Activity	Risk	Impact	Score	Rank	ORF Rank
Upload	Fails	Very high	86.99	1	1
Preview Alignment	Overload	High	8.58	2	2
ESYS Input-Share	Overload	High	3.46	3	3
Mapping Control	Overload	High	3.46	4	5
Repair / Adjustments	Overload	High	3.09	4	4
Preview Alignment	Wrong assessment	Medium	1.75	5	7
Cube-Tec Repair Server Input-Share	Overload	Medium	1.28	6	6
Mapping Control	Wrong assessment	Medium	1.28	7	7
Upload	Wrong parameters	Medium	0.64	8	8
ESYS Input-Share	Copy error	Low	0.19	9	9
TSM Retrieve	Wrong file selected	Low	0.14	10	11
TSM Retrieve	Retrieve fails	Low	0.14	11	10
Cube-Tec Repair Server Input-Share	Copy error	Low	0.03	12	12
Repair / Adjustments	Wrong assessment	Very low	0	13	13
Repair	Wrong mapping	Very low	0	13	14
Repair	Overload	Very low	0	13	14

evaluate risks and their actual impact even before or at an early stage of the implementation of a workflow. However, as discussed above, if the model is not configured with the right (and accurate) data, the results will not match with reality. Therefore, it is important to complete the risk management process, based on the Deming cycle, to capture monitoring information from the workflow executions to update both the risk information and simulation configuration to improve the accuracy.

VIII. Conclusion and Future Work

We have presented a Business Process Risk management framework (BPRisk) that allows users to manage workflow processes with regards to risk in order to reduce cost and increase the long-term value of digital media content. The framework is generic in nature, but has been discussed here in the context of digital preservation, where the objective is to avoid damage to the digital content and ensuring that the content can be accessed in the future.

The BPRisk framework combines workflow specification and risk management. It has been designed in accordance with a risk management process based on the Deming (PDCA) cycle and we have shown how it relates to the stages of the ISO 31000 risk methodology. Long-term digital preservation is threatened by format obsolescence, media degradation, and failures in the very people, processes and systems designed to keep the content safe and accessible. Therefore, investing in substantial planning and design is essential in order to prevent issues that may not be possible to rectify; rendering the content void. Further, key to the risk management process is continual improvement, i.e., risk management is not merely a static exercise performed at design time [13], but it is also imperative during process change.

A layered semantic risk model has been presented, which a) enables reasoning about threats in a workflow and b) assists end-users (who are typically not risk experts) by automatically suggesting relevant risks and respective controls for workflow activities. This helps the workflow designers specify more robust workflows, reducing the risk of causing irretrievable damage to the media content. Moreover, the framework helps workflow designers optimise workflows and improve cost-benefit by identifying (and addressing) key vulnerabilities by simulating workflow executions to estimate the impact of risk.

The simulation service in the BPRisk framework allows users to estimate the impact of risks before executing the workflow, which increases the chances of detecting issues rather than jeopardising the real media content. A workflow simulation provides users with information about *inter alia* the quantity of media content that may be affected by risk (and how), and the time and cost of dealing with risk (i.e., control and treatment). Therefore, organisational impacts can be derived and users may simulate different what-if scenarios in order to evaluate different workflow designs before proceeding with executing a particular workflow process on live material. What-if scenarios may, e.g., involve justifying reasons for putting in place certain control mechanisms by showing that the ROI is positive.

A prototype of the BPRisk framework has been developed in the DAVID project. To demonstrate the application of the framework and the value of the simulation results, we have reported on an evaluation scenario with the Austrian Broadcasting Corporation (ORF). The technical experts at ORF found the results to be almost identical to what they have observed by executing the workflow. Key benefits emphasised include: a) investing time in workflow planning and controlling risks in order to prevent issues, and b) justifying workflow designs and risk controls to decision makers.

Further research involves implementing mechanisms for automatically updating risk models and respective simulation configurations according to observed workflow execution data in order to improve the support for continual improvement of workflow processes.

Acknowledgements

This work has been carried out in the DAVID project, supported by the EC 7th Framework Programme (FP7-600827).

References

[1] V. Engen, G. Veres, S. Crowle, M. Bashevoy, P. Walland, and M. Hall-May, "A Semantic Risk Management Framework for Digital Audio-Visual Media Preservation," in The 10th International Conference on

Internet and Web Applications and Services (ICIW). IARIA, June 2015, pp. 81–87.

[2] EC FP7 DAVID Project, "Digital AV Media Damage Prevention and Repair," http://david-preservation.eu/, [retrieved: 2016.02.29].

[3] Object Management Group, "Business Process Model and Notation (BPMN) Version 2.0," http://www.omg.org/spec/BPMN/2.0/PDF/, [retrieved: 2016.02.29].

[4] Department of Special Collections, Donald C. Davidson Library, University of California, "Cylinder Preservation and Digitization Project," http://cylinders.library.ucsb.edu/, [retrieved: 2016.02.29].

[5] V. Engen, G. Veres, M. Hall-May, J.-H. Chenot, C. Bauer, W. Bailer, M. Höffernig, and J. Houpert, "Final IT Strategies & Risk Framework," EC FP7 DAVID Project, Tech. Rep. D3.3, 2014, available online http://david-preservation.eu/wp-content/uploads/2013/01/DAVID-D3.3-Final-IT-Strategies-Risk-Framework.pdf [retrieved: 2016.02.29].

[6] M. Addis, R. Wright, and R. Weerakkody, "Digital Preservation Strategies: The Cost of Risk of Loss," SMPTE Motion Imaging Journal, vol. 120, no. 1, 2011, pp. 16–23.

[7] J.-H. Chenot and C. Bauer, "Data damage and its consequences on usability," EC FP7 DAVID Project, Tech. Rep. D2.1, 2013, available online http://david-preservation.eu/wp-content/uploads/2013/10/DAVID-D2-1-INA-WP2-DamageAssessment_v1-20.pdf [retrieved: 2016.02.29].

[8] D. Rosenthal, "Format Obsolescence: Assessing the Threat and the Defenses," Library Hi Tech, vol. 28, no. 2, 2010, pp. 195–210.

[9] M. Addis, "8K Traffic Jam Ahead," PrestoCentre Blog, April 2013, available online: https://www.prestocentre.org/blog/8k-traffic-jam-ahead [retrieved: 2016.02.29].

[10] ISO/IEC, 31000:2009 Risk management - Principles and guidelines, ISO Std., 2009.

[11] D. Green, K. Albrecht, and et al, "The NINCH Guide to Good Practice in the Digital Representation and Management of Cultural Heritage Materials," National Initiative for a Networked Cultural Heritage, Tech. Rep., 2003, available online: http://www.ninch.org/guide.pdf [retrieved: 2016.02.29].

[12] M. Hall-May, B. Arab-Zavar, J. Houpert, C. Tiensch, H. Fassold, and V. Engen, "Analysis of loss modes in preservation systems," EC FP7 DAVID Project, Tech. Rep. D2.2, 2014, available online http://david-preservation.eu/wp-content/uploads/2013/01/DAVID-D2.2-Analysis-of-Loss-Modes-in-Preservation-Systems.pdf [retrieved: 2016.02.29].

[13] S. Suriadi, B. Weiß, A. Winkelmann, A. Hofstede, M. Adams, R. Conforti, C. Fidge, M. La Rosa, C. Ouyang, A. Pika, M. Rosemann, and M. Wynn, "Current Research in Risk-Aware Business Process Management - Overview, Comparison, and Gap Analysis," BPM Center, Tech. Rep. BPM-12-13, 2012.

[14] M. Rosemann and M. zur Muehlen, "Integrating Risks in Business Process Models," in ACIS Proceedings, 2005.

[15] A. Sienou, E. Lamine, A. Karduck, and H. Pingaud, "Conceptual Model of Risk: Towards a Risk Modelling Language," in Web Information Systems Engineering, ser. LNCS 4832, 2007, pp. 118–129.

[16] R. Conforti, G. Fortino, and A. t. M. La Rosa, "History-Aware, Real-Time Risk Detection in Business Processes," in On the Move to Meaningful Internet Systems, ser. LNCS. Springer, 2011, vol. 7044, pp. 100–118.

[17] X. Bai, R. Krishnan, R. Padman, and H. Wang, "On Risk Management with Information Flows in Business Processes," Information Systems Research, vol. 24, no. 3, 2013, pp. 731–749.

[18] M. Addis, M. Jacyno, M. H. Hall-May, and S. Phillips, "Tools for Quantitative Comparison of Preservation Strategies," EC FP7 PrestoPRIME Project, Tech. Rep. D2.1.4, 2012, available online: http://eprints.soton.ac.uk/349290/ [retrieved: 2016.02.29].

[19] V. Lloyd, ITIL Continual Service Improvement – 2011 Edition. The Stationary Office, 2011, iSBN: 9780113313082.

[20] Signavio GmbH, "Signavio Decision Manager," http://www.signavio.com/products/decision-manager/, [retrieved: 2016.02.29].

[21] JBoss, "jBPM," http://www.jboss.org/jbpm, [retrieved: 2016.02.29].

[22] L. Richardson and S. Ruby, RESTful Web Services, 1st ed. O'Reilly, May 2007.

[23] Pivotal Software, "Spring," https://spring.io/, [retrieved: 2016.02.29].

[24] G. E. Krasner and S. T. Pope, "A cookbook for using the model-view controller user interface paradigm in Smalltalk-80," Journal of Object-Oriented Programming, vol. 1, no. 3, Aug/Sept 1988, pp. 26–49.

[25] Activiti, "Activiti BPM Platform," http://activiti.org/, [retrieved: 2016.02.29].

[26] Mathworks, "Simulink," http://uk.mathworks.com/products/simulink/, [retrieved: 2016.02.29].

[27] L. Moreau and P. Missier, "PROV-DM: The PROV Data Model," W3C Recommendation, 2013, available online: http://www.w3.org/TR/2013/REC-prov-dm-20130430/ [retrieved: 2016.02.29].

[28] M. Surridge, A. Chakravarthy, M. Hall-May, X. Chen, B. Nasser, and R. Nossal, "SERSCIS: Semantic Modelling of Dynamic, Multi-Stakeholder Systems," in 2nd SESAR Innovations Days, 2012.

[29] F. Bellard, "FFmpeg," https://www.ffmpeg.org/, [retrieved: 2016.02.29].

[30] W3C, "RDF 1.1 Turtle," W3C Recommendation, 2014, available online: https://www.w3.org/TR/turtle [retrieved: 2016.02.29].

[31] ——, "SPARQL Inferencing Notation (SPIN)," W3C Submission, 2011, available online: http://www.w3.org/Submission/2011/02/ [retrieved: 2016.02.29].

[32] S. Vermaaten, B. Lavoie, and P. Caplan, "Identifying Threats to Successful Digital Preservation: the SPOT Model for Risk Assessment," D-Lib Magazine, vol. 18, no. 9/10, 2012.

[33] Cube-Tec International, "Cubeworkflow," https://www.cube-tec.com/en-uk/products/workflow/cube-workflow/cube-workflow-20, 2016, [retrieved: 2016.02.29].

[34] ISO/IEC, ISO/IEC 13818-1:2000 Information technology – Generic coding of moving pictures and associated audio information: Systems, ISO Std., Dec 2000.

[35] SMTPE, "For Television – Type D-10 Stream Specifications – MPEG-2 4:2:2P @ ML for 525/60 and 625/50," SMPTE ST 356:2001, Aug 2001, pp. 1–7.

An Investigation of a Factor that Affects the Usage of Unsounded Code Strings at the End of Japanese, English, Spanish, Portuguese, and French Tweets

Yasuhiko Watanabe, Kunihiro Nakajima, Haruka Morimoto, Ryo Nishimura, and Yoshihiro Okada
Ryukoku University
Seta, Otsu, Shiga, Japan
Email: watanabe@rins.ryukoku.ac.jp, t13m071@mail.ryukoku.ac.jp, t13m076@mail.ryukoku.ac.jp,
r_nishimura@afc.ryukoku.ac.jp, okada@rins.ryukoku.ac.jp

***Abstract*—In this study, we compare Japanese, English, Spanish, Portuguese, and French tweets submitted to Twitter and discuss how we use unsounded code strings at the end of online messages. We first define unsounded codes and unsounded code strings. Next, we compare and discuss the usage of unsounded code strings at the end of tweets, especially, to general public and particular persons. Finally, we show the receiver of a tweet, whether general public or a particular person, is a factor that affects the usage of unsounded code strings at the end of Japanese tweets, but not English, Spanish, Portuguese, and French tweets. Specifically, Japanese speakers use unsounded code strings at the end of tweets more frequently to particular persons than to general public while English, Spanish, Portuguese, and French speakers do not.**

Keywords–unsounded code string; Twitter; general public; particular persons; non verbal communication.

I. INTRODUCTION

Many of us think that it is easy to understand the meanings of non verbal expressions in online messages even if different language speakers generate them. However, the usage differs between different language speakers and the difference is at risk of bringing unnecessary frictions between them. As a result, it is important to consider the difference, especially, in multilingual computer-mediated communication (CMC) systems. To solve this problem, we showed the usage of unsounded code strings, one kind of non verbal expression, differs between Japanese and English speakers and discuss a factor that affects the usage of them [1].

We often find consecutive unsounded marks and characters are used at the end of online messages, such as mails, chattings, and tweets in Twitter.

(exp 1) I'm freezing!!!!
(exp 2) *@ryuuuuuuuu_2012 soushita hou ga iiyo.....*
(@ryuuuuuuuu_2012 you had better do it.....)

(exp 1) and (exp 2) are tweets submitted to Twitter. (exp 1) was submitted by an user who chose English as his/her language for tweets. On the other hand, (exp 2) was submitted by an user who chose Japanese as his/her language for tweets. Both (exp 1) and (exp 2) have consecutive unsounded marks at the end of them. These unsounded marks are used for smooth communication. The submitter of (exp 1) is thought to use the three consecutive exclamation marks for expressing his/her impression strongly. On the other hand, the submitter of (exp 2) is thought to use the seven consecutive periods for expressing his/her opinion softly. In this study, we define unsounded marks and characters as *unsounded codes*. Furthermore, we define three or more consecutive unsounded codes as a *unsounded code string*. For example, in Twitter, 14 % of Japanese tweets and 10 % of English tweets have unsounded code strings at the end of them. Although unsounded code strings are popular, there are few studies on them. As a result, in this study, we investigate how we use unsounded code strings at the end of tweets in Twitter. Especially, we compare Japanese, English, Spanish, Portuguese, and French tweets in Twitter and discuss a factor that affects the usage of unsounded code strings at the end of tweets. The results of this study will give us a chance to understand the usage of unsounded code strings and improve multilingual CMC systems.

The rest of this paper is organized as follows: In Section II, we survey the related works. In Section III, we define unsounded code strings and describes how they are used at the end of tweets in Twitter. Finally, in Section IV, we present our conclusions.

II. RELATED WORKS

There are a considerable number of studies comparing speakers of various languages from various viewpoints, such as, pragmatics, cognitive science, and so on. These studies can be classified into two types:

- studies comparing native speakers of one language to non-native speakers of the same language
- studies comparing native speakers of one language to native speakers of other language directly

This study is classified into the latter. It is because we compare unsounded code strings in one language tweets to those in other language tweets directly.

In pragmatics, a considerable number of studies have been made on interlanguage speech acts, such as, expressing compliments [2], apologies [3], gratitude [4], politeness [5], and refusals [6]. In these studies, native speakers of one language were compared to non-native speakers of the same language. Also, there are a considerable number of studies comparing pauses and backchannels of native speakers to those of non-native speakers. Backchannels are listener's responses, such as "uh-huh" and "yeah", given while someone else is talking, to show an interest, attention, or willingness to keep listening. Deschamps investigated how French learners of English made pauses in their English speeches [7]. Bilá and Džambová investigated the function of silent pauses in native and non-native speakers of English and German [8]. Ishizaki investigated how English, French, Chinese and Korean learners of Japanese and native Japanese speakers made pauses in their Japanese speeches [9]. Tavakoli reported that the location of pauses is important in comparisons of native speakers and

foreign learners [10]. Okazawa reported that native Japanese speakers paused roughly twice as often in their English utterances than in their Japanese utterances [11]. LoCastro reported that Japanese speakers often feel uncomfortable when speaking English because they are unable to use the appropriate backchannels [12]. From the viewpoint of cognitive science, Tera et al. compared and analyzed reading processes of native Japanese speakers and foreign learners of Japanese [13].

On the other hand, there are also a considerable number of studies comparing native speakers of one language to native speakers of other language directly. Especially, many studies have been made on backchannels. It is because backchannels are found in various languages. Maynard showed that backchannel phenomena for Japanese and English differ in terms of type, frequency, and context [14]. Miller reported that Japanese speakers use backchannels more frequently than English speakers [15]. However, in most of previous studies, the frequency of backchannels were observed in various situations while little is known about factors that affect the frequency of backchannels. Chen pointed out that it is important to investigate factors that affect the frequency of backchannels and took White's work [16] for example [17]. White reported that Americans used backchannels more frequently in conversations with Japanese than in conversations with other Americans [16]. In other words, the conversation partner, whether American or Japanese, is a factor that affects the frequency of Americans' backchannels. As a result, when we compare different language speakers, it is important to investigate factors providing different responses between them. In this study, we investigate a factor that affects the usage of unsounded code strings at the end of tweets.

III. Unsounded code strings at the end of Japanese, English, Spanish, Portuguese, and French tweets in Twitter

In this section, we compare and discuss the usage of unsounded code strings at the end of Japanese, English, Spanish, Portuguese, and French tweets. In Section III-A, we define unsounded code strings and show how they are used at the end of tweets. In Section III-B, we show the investigation object of this study. Finally, in Section III-C, we compare Japanese, English, Spanish, Portuguese, and French tweets and discuss a factor that affects the usage of unsounded code strings at the end of tweets.

A. The definition of an unsounded code string

First, we define unsounded codes and unsounded code strings. In this study, we define that an unsounded code string is three or more consecutive unsounded codes. In this study, unsounded codes in English, Spanish, Portuguese, and French text are limited to

- punctuation marks (e.g., !#$%&.,:;<=>?@ (){}).

On the other hand, unsounded codes in Japanese text are limited to the following marks and characters:

- punctuation marks,
- Greek characters,
- Cyrillic characters, and
- ruled lines.

It is because these marks and characters are generally unsounded when they are used at the end of Japanese sentences.

Next, we show how unsounded code strings are used at the end of tweets. Twitter users often use unsounded code strings in order to enable anyone to understand their tweets clearly and avoid unnecessary frictions with others.

(exp 3) *kadai owattaaaaaaaaaaaa!!!!!!!!!!* (I got my homework doooooooooooone!!!!!!!!!!)
(exp 4) @jayne_hurley looks amazing !!!
(exp 5) WEIRDEST dream last night omg...

For example, the submitters used exclamation marks consecutively at the end of (exp 3) and (exp 4) for expressing their feelings strongly. On the other hand, the submitter used periods consecutively at the end of (exp 5) for expressing his/her impression softly.

We may note that some submitters use unsounded code strings not for expressing their feelings or intentions clearly. For example, Twitter users are prohibited to post the same tweets repeatedly. To avoid this constraint, some users use unsounded code strings. For example, (exp 6), (exp 7), and (exp 8) were posted consecutively in a few seconds to a particular user, *ssuzuki16*, beyond the limit of repeated submission.

(exp 6) *@ssuzuki16 yo...*
(exp 7) *@ssuzuki16 yo....*
(exp 8) *@ssuzuki16 yo.....*

B. The investigation object

We obtained tweets by using the streaming API [18]. However, the streaming API allows us to obtain only 1% of all public streamed tweets because of API restriction. We used the streaming API and obtained 56,483,681 tweets in three weeks in November and December 2012. The obtained tweets included the following tweets.

- 31,253,241 tweets submitted by users who chose English as their language for tweets. In this study, we call these tweets *English tweets*.
- 7,254,205 tweets submitted by users who chose Spanish as their language for tweets. In this study, we call these tweets *Spanish tweets*.
- 7,085,267 tweets submitted by users who chose Japanese as their language for tweets. In this study, we call these tweets *Japanese tweets*.
- 3,101,308 tweets submitted by users who chose Portuguese as their language for tweets. In this study, we call these tweets *Portuguese tweets*.
- 1,075,959 tweets submitted by users who chose French as their language for tweets. In this study, we call these tweets *French tweets*.

Only these five language tweets were more than 1,000,000 tweets. Other language tweets (e.g., German and Italian tweets) were less than 1,000,000 tweets. Figure 1 shows the percentage of English, Spanish, Japanese, Portuguese, and French tweets in the obtained tweets. These tweets can be classified into three types:

- reply
  A reply is submitted to a particular person (Figure 2). It contains "@username" in the body of the tweet. For example, (exp 2) and (exp 4) are replies.

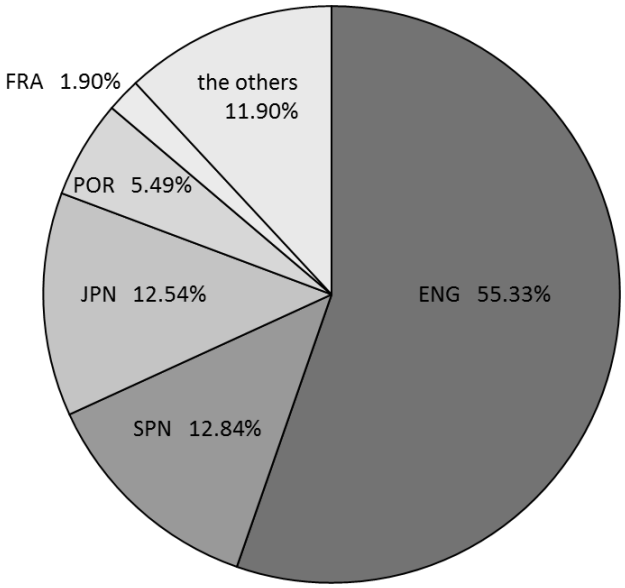


Figure 1. The percentage of English, Spanish, Japanese, Portuguese, and French tweets in the obtained 56,483,681 tweets (in three weeks in November and December 2012).

- retweet
  A retweet is a reply to a tweet that includes the original tweet.
- normal tweet
  A normal tweet is neither reply nor retweet. For example, (exp 1), (exp 3), and (exp 5) are normal tweets. Twitter users generally submit their tweets to general public. As a result, most of normal tweets are submitted to general public (Figure 3).

Figure 4 shows the percentages of normal tweets, replies, and retweets in the obtained Japanese, English, Spanish, Portuguese, and French tweets. From the obtained Japanese tweets, we extracted 966,187 Japanese tweets that have unsounded code strings at the end of them. These 966,187 Japanese tweets are 13.64% of all the Japanese tweets. On the other hand, from the obtained English tweets, we extracted 3,270,821 English tweets that have unsounded code strings at the end of them. These 3,270,821 English tweets are 10.47% of all the English tweets. Figure 5 shows the percentages of the obtained Japanese, English, Spanish, Portuguese, and French tweets that have unsounded code strings at the end of them. Furthermore, Figure 6 shows the percentages of normal tweets, replies, and retweets in the obtained Japanese, English, Spanish, Portuguese, and French tweets that have unsounded code strings at the end of them.

In this study, we do not discuss unsounded code strings at the end of retweets. It is because, messages in retweets are created not by submitters, but by other users. As a result, retweets are inadequate to investigate how we use unsounded code strings at the end of online messages.

In this study, we compare unsounded code strings at the end of (1) normal tweets and (2) replies. It is because we intend to compare and discuss the usage of unsounded code strings at the end of tweets to general public and particular persons. As mentioned, normal tweets are generally submitted to general public. On the other hand, each reply is submitted to a particular person.

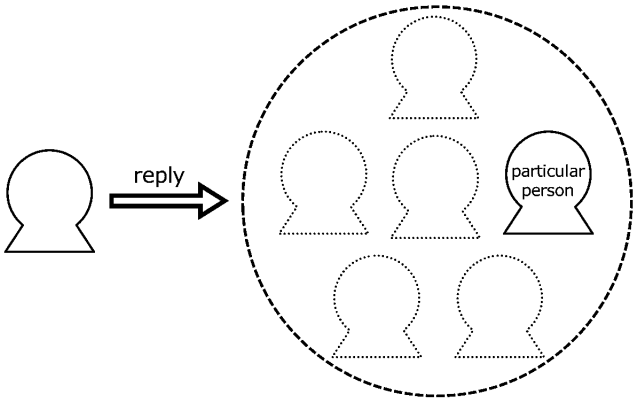


Figure 2. A reply is submitted to a particular person.

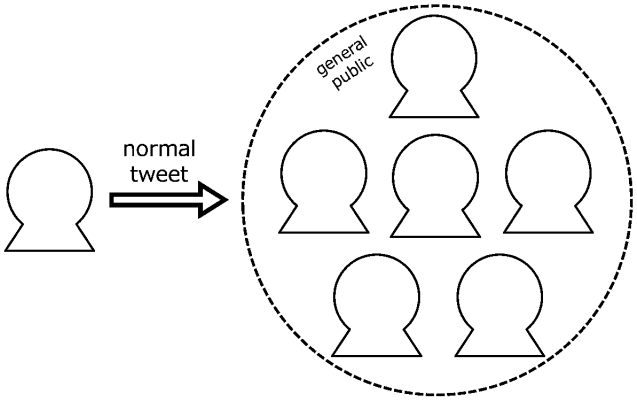


Figure 3. A normal tweet is submitted to general public.

C. The comparison of the usage of unsounded code strings at the end of Japanese and English tweets

Japanese tweets differs greatly from English, Spanish, Portuguese, and French tweets. For example, Kanji characters, Hiragana letters, and Katakana leters are mainly used in Japanese tweets while alphabetical letters are mainly used in English, Spanish, Portuguese, and French tweets. In this section, we compare Japanese tweets and English tweets because English tweets are more than Spanish, Portuguese, and French tweets.

Figure 7 and Figure 8 show the cumulative relative frequency distribution of

- the length of all the Japanese and English tweets (excluding retweets),
- the length of Japanese and English tweets (excluding retweets) that have unsounded code strings at the end of them, and
- the length of unsounded code strings at the end of Japanese and English tweets (excluding retweets).

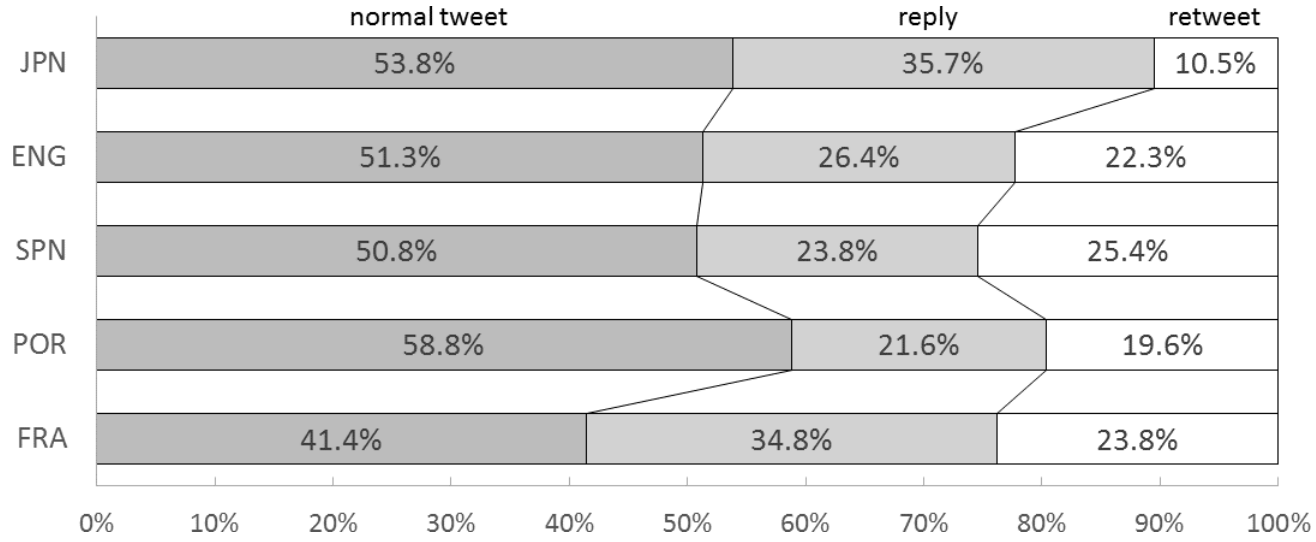


Figure 4. The percentages of normal tweets, replies, and retweets in the obtained Japanese, English, Spanish, Portuguese, and French tweets.

Figure 5. The percentages of Japanese, English, Spanish, Portuguese, and French tweets that have unsounded code strings at the end of them.

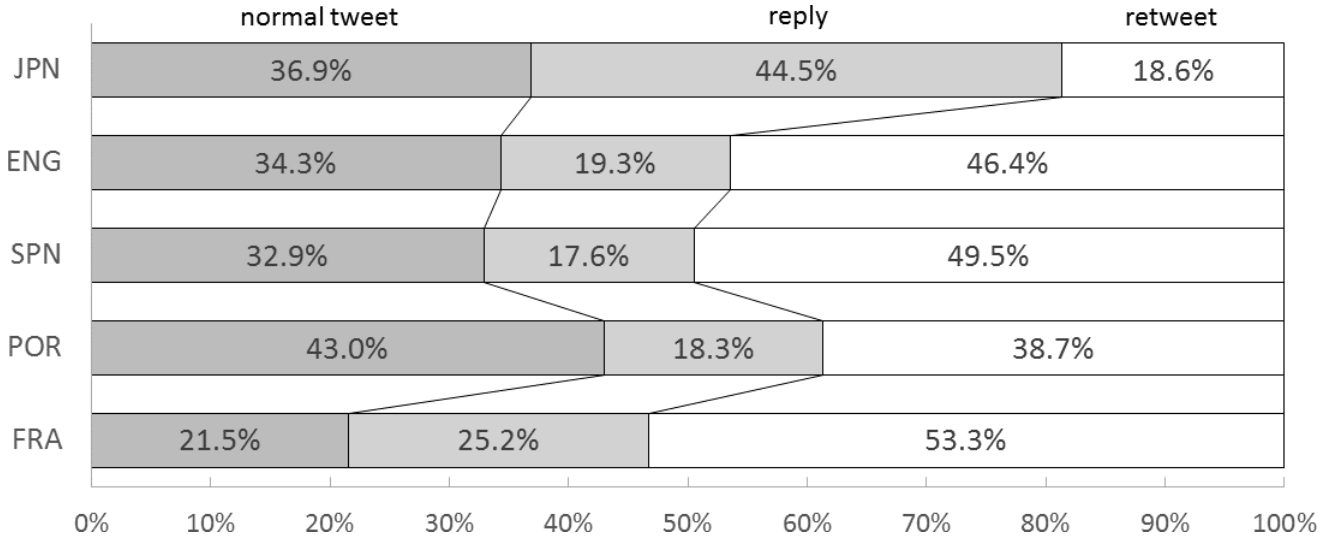


Figure 6. The percentages of normal tweets, replies, and retweets in the obtained Japanese, English, Spanish, Portuguese, and French tweets that have unsounded code strings at the end of them.

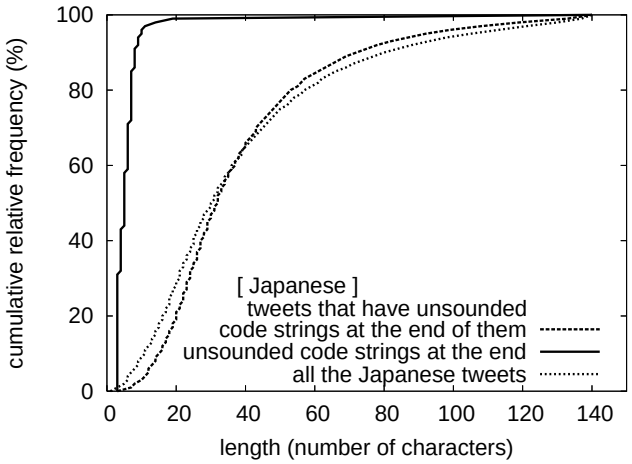


Figure 7. The cumulative relative frequency distribution of the length of (1) all the Japanese tweets, (2) Japanese tweets that have unsounded code strings at the end of them, and (3) unsounded code strings at the end of them.

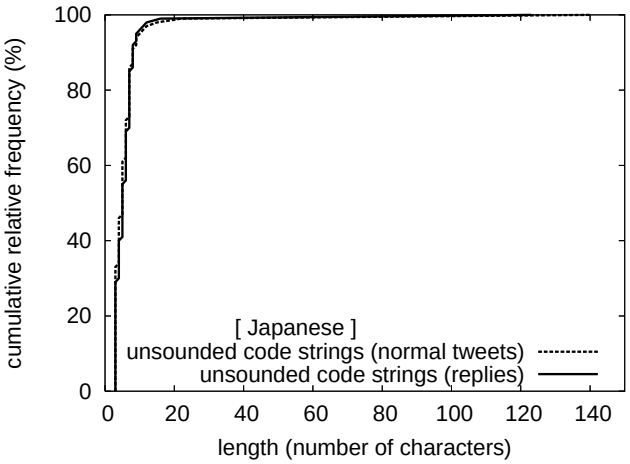


Figure 9. The cumulative relative frequency distribution of the length of unsounded code strings at the end of Japanese normal tweets and replies.

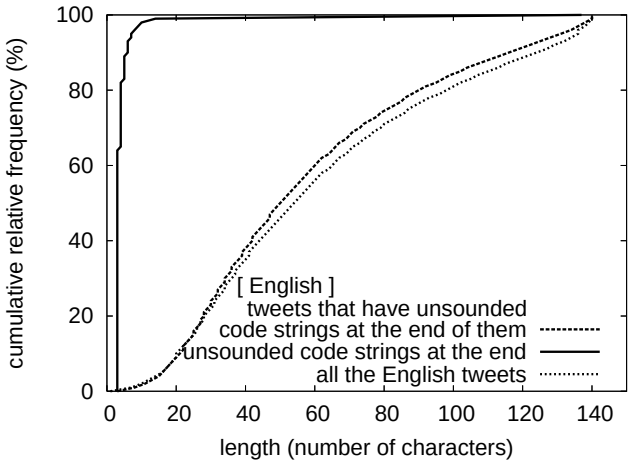


Figure 8. The cumulative relative frequency distribution of the length of (1) all the English tweets, (2) English tweets that have unsounded code strings at the end of them, and (3) unsounded code strings at the end of them.

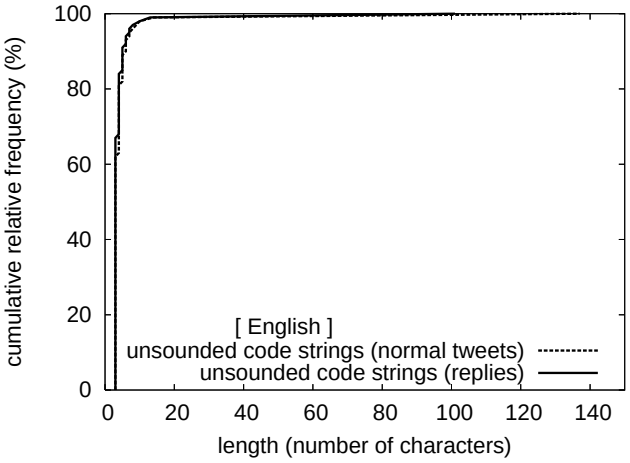


Figure 10. The cumulative relative frequency distribution of the length of unsounded code strings at the end of English normal tweets and replies.

As shown in Figure 7 and Figure 8, the distribution of the length of Japanese tweets shifts to shorter ranges than the length of English tweets. On the other hand, the distribution of the length of unsounded code strings at the end of Japanese tweets shifts to longer ranges than the length of those at the end of English tweets.

Next, we compare the length of unsounded code strings at the end of normal tweets and replies. Figure 9 and Figure 10 show the cumulative relative frequency distribution of the length of unsounded code strings at the end of

- Japanese normal tweets and replies, and
- English normal tweets and replies.

As shown in Figure 9 and Figure 10, the length of unsounded code strings at the end of Japanese and English normal tweets have a similar distribution pattern to those of Japanese and English replies, respectively. As a result, it may be said that the length of unsounded code strings at the end of tweets are less affected by whether the tweets are normal tweets or replies.

Next, we compare the length of normal tweets and replies that have unsounded code strings at the end of them. Figure 11 and Figure 12 show the cumulative relative frequency distribution of

- the length of all the Japanese and English normal tweets, and
- the length of Japanese and English normal tweets that have unsounded code strings at the end of them.

On the other hand, Figure 13 and Figure 14 show the cumulative relative frequency distribution of

- the length of all the Japanese and English replies, and
- the length of Japanese and English replies that have unsounded code strings at the end of them.

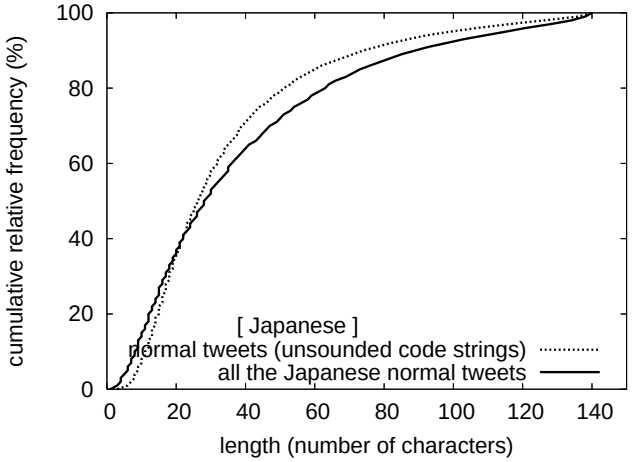


Figure 11. The cumulative relative frequency distribution of the length of all the Japanese normal tweets and Japanese normal tweets that have unsounded code strings at the end of them.

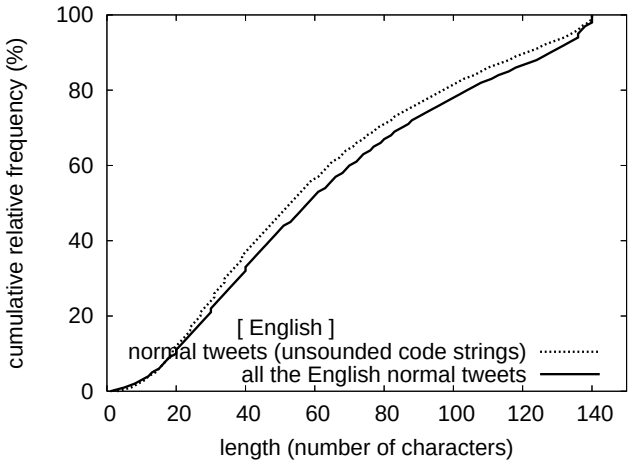


Figure 12. The cumulative relative frequency distribution of the length of all the English normal tweets and English normal tweets that have unsounded code strings at the end of them.

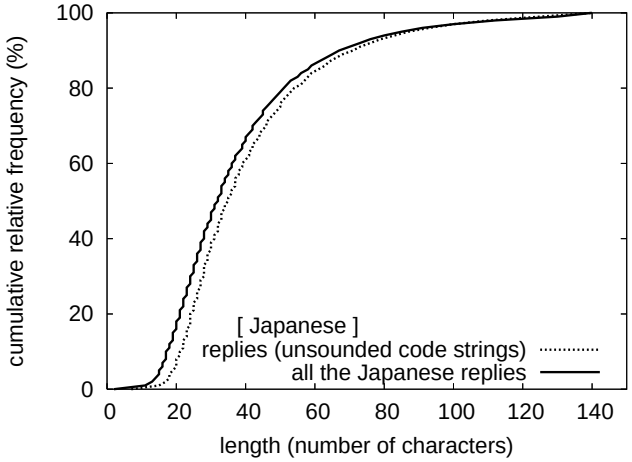


Figure 13. The cumulative relative frequency distribution of the length of all the Japanese replies and Japanese replies that have unsounded code strings at the end of them.

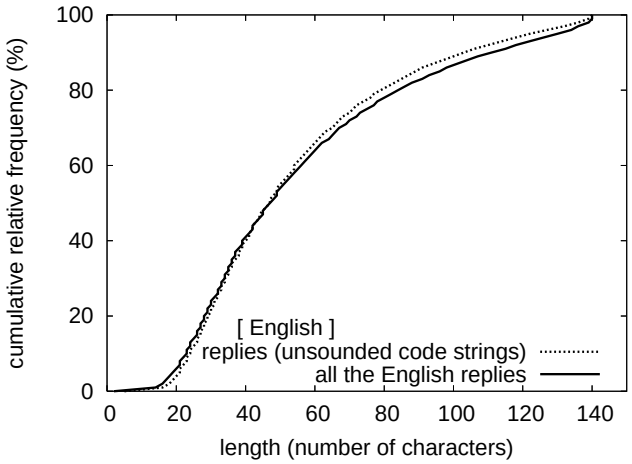


Figure 14. The cumulative relative frequency distribution of the length of all the English replies and English replies that have unsounded code strings at the end of them.

As shown in Figures 11–14, only the distribution of the length of Japanese replies that have unsounded code strings at the end of them shifts to longer ranges than the length of all the Japanese replies. On the other hand, the distribution of the length of the other tweets (Japanese normal tweets, English normal tweets and replies) that have unsounded code strings at the end of them do not shift to longer ranges when they are longer than 30 characters. It may be said that the length of Japanese tweets that have unsounded code strings at the end of them are affected by whether the tweets are normal tweets or replies. On the other hand, the length of English tweets that have unsounded code strings at the end of them are not affected.

Next, we investigate that the percentages of tweets that have unsounded code strings at the end of them are affected by whether the tweets are normal tweets or replies. Figure 15 shows the percentages of Japanese normal tweets and replies that have unsounded code strings at the end of them. As shown in Figure 15, 9.4 % of Japanese normal tweets have unsounded code strings at the end of them while 17.0 % of Japanese replies have unsounded code strings at the end of them. As a result, the percentages of Japanese normal tweets and replies that have unsounded code strings at the end of them differ considerably from each other. In other words, Japanese replies have unsounded code strings at the end of them more frequently than Japanese normal tweets. On the other hand, Figure 16 shows the percentages of English normal tweets and replies that have unsounded code strings at the end of them. As shown in Figure 16, 7.0 % of English normal tweets have unsounded code strings at the end of them while 7.6 % of English replies have unsounded code strings at the end of them. As a result, the percentages of English normal tweets and replies that have unsounded code strings at the end of them differ little from each other. In addition, the percentages

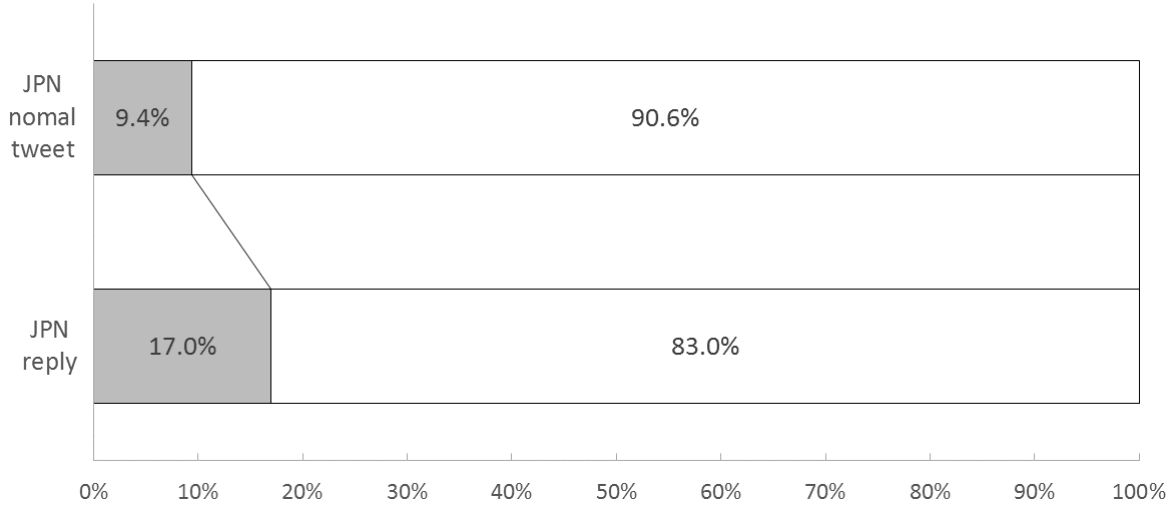


Figure 15. The percentages of Japanese normal tweets and replies that have unsounded code strings at the end of them

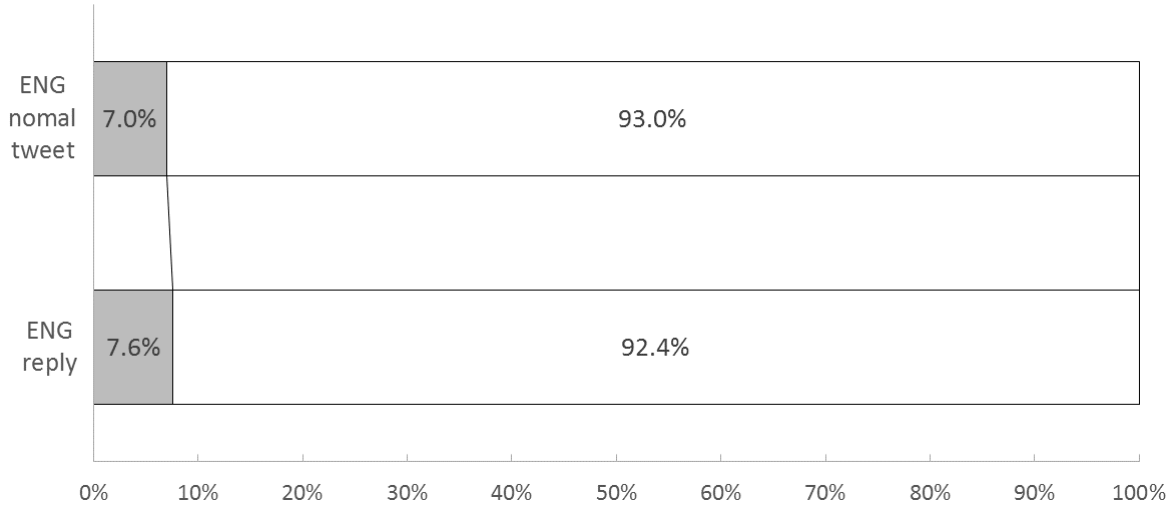


Figure 16. The percentages of English normal tweets and replies that have unsounded code strings at the end of them

of English normal tweets and Japanese normal tweets that have unsounded code strings at the end of them differ little from each other. From these points, it may be said that Japanese speakers use unsounded code strings at the end of tweets more frequently to particular persons than to general public while English speakers do not. In other words, the receiver of a tweet, whether general public or a particular person, is a factor that affects the usage of unsounded code strings for Japanese speakers, however, not for English speakers.

D. The investigation of the usage of unsounded code strings at the end of Spanish, Portuguese, and French tweets

We found that Japanese speakers use unsounded code strings at the end of tweets more frequently to particular persons than to general public while English speakers do not. However, it is not clear whether this phenomenon is specific for Japanese speakers. To solve this problem, we investigate that the percentages of Spanish, Portuguese, and French tweets that have unsounded code strings at the end of them are affected by whether the tweets are normal tweets or replies. Figure 17 shows the percentages of Spanish normal tweets and replies that have unsounded code strings at the end of them. As shown in Figure 17, 7.5 % of Spanish normal tweets have unsounded code strings at the end of them while 8.6 % of Spanish replies have unsounded code strings at the end of them. Figure 18 shows the percentages of Portuguese normal tweets and replies that have unsounded code strings at the end of them. As shown in Figure 18, 6.4 % of Portuguese normal tweets have unsounded code strings at the end of them while 7.4 % of Portuguese replies have unsounded code strings at the end of them. Figure 19 shows the percentages of French normal tweets and replies that have unsounded code strings at the end of them. As shown in Figure 19, 5.0 % of French normal tweets have unsounded code strings at the end of them while 7.0 % of French replies have unsounded code strings at the end of them. As a result, as in the case of English tweets, the percentages of Spanish, Portuguese, and French normal tweets that have unsounded code strings at the end of them differ little

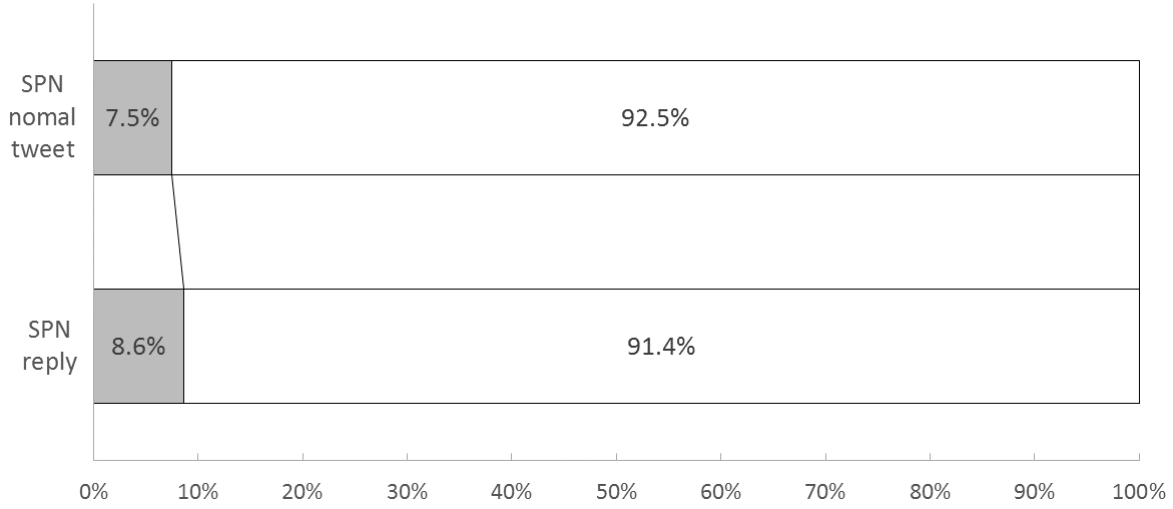


Figure 17. The percentages of Spanish normal tweets and replies that have unsounded code strings at the end of them

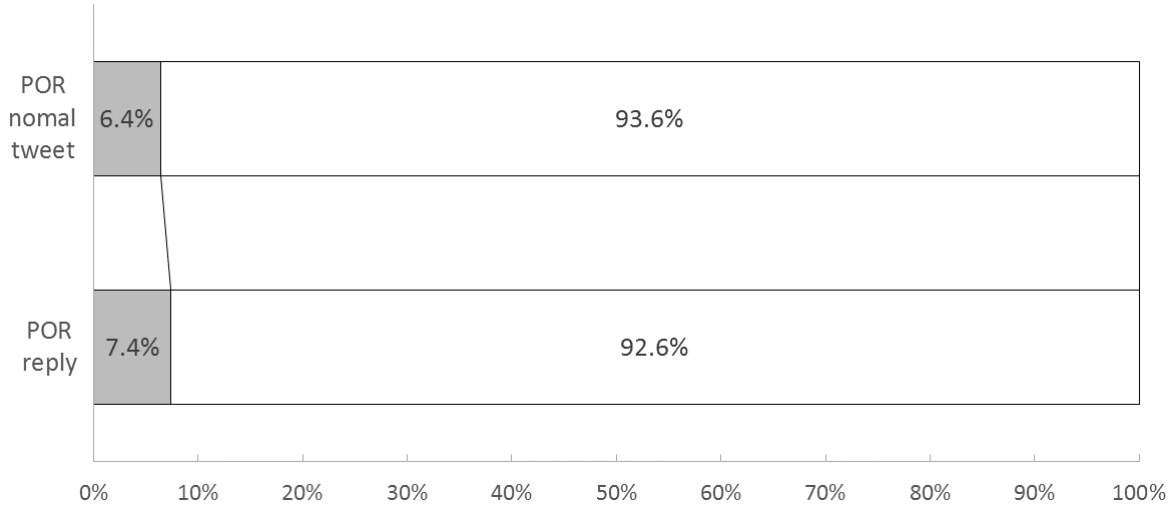


Figure 18. The percentages of Portuguese normal tweets and replies that have unsounded code strings at the end of them

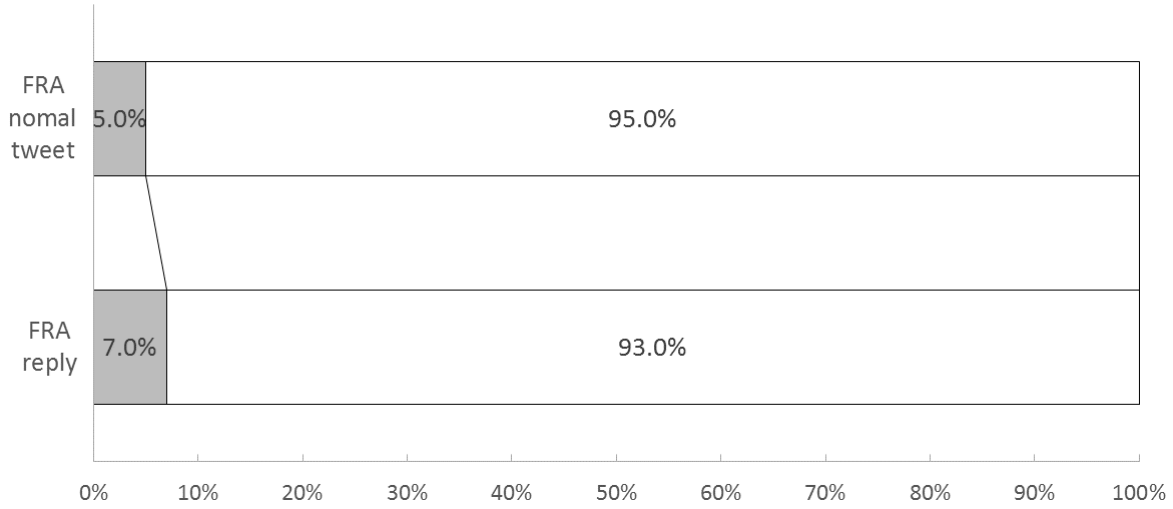


Figure 19. The percentages of French normal tweets and replies that have unsounded code strings at the end of them

from those of their replies that have unsounded code strings at the end of them, respectively. From these points, it may be said that Japanese speakers use unsounded code strings at the end of tweets more frequently to particular persons than to general public while English, Spanish, Portuguese, and French speakers do not.

Next, we discuss why Japanese speakers differ from other language speakers. We think that the difference is caused by whether speakers change their expressions according to listeners. In order to express speakers' attitudes to listeners clearly, Japanese speakers generally change their expressions according to listener's age, gender, position, feeling, and so on. These expressions in Japanese are studied as a topic of modality expressions [19] [20]. Masuoka reported that Japanese speakers use modality expressions more frequently than other language speakers [21].

(exp 9) *gakkou ni ike* (Go to the school)
(exp 10) *gakkou ni ike yo* (Go to the school, OK?)

In both (exp 9) and (exp 10), the speakers order the listeners to go to the school. However, the speaker of (exp 9) simply orders the listener to go to the school. On the other hand, the speaker of (exp 10) uses a modality expression "*yo*" and shows that he/she understands listener's feeling: the listener does not want to go to the school. Because of the modality expression, the listener can receive (exp 10) more softly than (exp 9). The point is that modality expression "*yo*" of (exp 10) is similar to unsounded code string "..." of (exp 11) in the meaning.

(exp 11) *gakkou ni ike...* (Go to the school...)

It is likely that many Japanese speakers use unsounded code strings at the end of Japanese sentences as one kind of modality expressions. As a result, we think that Japanese speakers change expressions at the end of tweets according to listener, whether general public and particular persons, while other language speakers do not.

There is another point of view: the difference of the usage of unsounded code strings is caused by culture. Nisbett reported that how we think is influenced by culture [22]. He showed that people who grow up in East Asia pay more attention to context and background than people who grow up in the West. The theory is that East Asians grow up learning to pay attention to context because cultural norms in East Asia emphasize relationships and groups. On the other hand, Westerners grow up learning to pay more attention to focal objects than context because Western society is more individualistic than East Asia society. From this point of view, it may be said, because of cultural norms in Japan, Japanese speakers pay more attention to relationships with listeners than English, Spanish, Portuguese, and French speakers. To discuss this matter, it is important to introduce geo-location information associated with Tweets into our investigation. By using geo-location information, we can distinguish and investigate native speakers that speak the same language but belong to the different cultures, for example, British and other native English speakers.

We may note that Internet evolution gives us a new communication media from individuals to general public and we adapt to it rapidly. Furthermore, the adaptations differ depending on languages. The results of this study will give us a chance to understand the usage of unsounded code strings and improve multilingual CMC systems.

IV. Conclusion

Unsounded code strings, in other words, consecutive unsounded marks and characters are frequently used at the end of online messages. However, there were few studies on them. In this study, we investigated unsounded code strings at the end of Japanese, English, Spanish, Portuguese, and French tweets in Twitter. Then, we showed that Japanese speakers use unsounded code strings at the end of tweets more frequently to particular persons than to general public while English, Spanish, Portuguese, and French speakers do not. It may be said that the receiver of a tweet, whether general public or a particular person, is a factor that affects the usage of unsounded code strings for Japanese speakers, however, not for the other language speakers. It is because, we think, Japanese speakers generally change their expressions according to listener's age, gender, position, feeling, and so on while the other language speakers do not. In order to discuss whether this phenomenon is specific for Japanese speakers, we intend to analyze tweets in various languages, especially, languages in East Asia. Furthermore, we intend to introduce geo-location information into our study and examine whether expressions in tweets are affected by areas. The results of this study will give us a chance to understand the usage of unsounded code strings and improve multilingual CMC systems.

References

[1] Y. Watanabe, K. Nakajima, H. Morimoto, R. Nishimura, and Y. Okada, "An investigation of a factor that affects the usage of unsounded code strings at the end of japanese and english tweets," in Proceedings of the Seventh International Conference on Evolving Internet (INTERNET 2015), Oct 2015, pp. 50–55. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=internet_2015_2_40_40038 [accessed: 2016-6-1]

[2] N. Wolfson, "The social dynamics of native and nonnative variation in complimenting behavior," in The Dynamic Interlanguage: Empirical Studies in Second Language Variation. Springer US, 1989, pp. 219–236.

[3] M. L. Bergman and G. Kasper, "Perception and performance in native and nonnative apology," in Interlanguage pragmatics. Oxford University Press, 1993, pp. 82–107.

[4] M. Eisenstein and J. Bodman, "Expressing gratitude in American English," in Interlanguage pragmatics. Oxford University Press, 1993, pp. 64–81.

[5] S. Tanaka and S. Kawade, "Politeness strategies and second language acquisition," Studies in Second Language Acquisition, vol. 5, no. 1, 1982, pp. 18–33.

[6] L. M. Beebe, T. Takahashi, and R. Uliss-Weltz, "Pragmatic transfer in ESL refusals," in Developing communicative competence in a second language. Newbury House Publishers, 1990, pp. 55–73.

[7] A. Deschamps, "The syntactical distribution of pauses in English spoken as a second language by French students," in Temporal variables in speech. De Gruyter Mouton Publishers, 1980, pp. 255–262.

[8] M. Bilá and A. Džambová, "A preliminary study on the function of silent pauses in L1 and L2 speakers of English and German," Brno Studies in English, vol. 37, no. 1, 2011, pp. 21–39.

[9] A. Ishizaki, "How does a learner leave a pause when reading Japanese aloud?: A comparison of English, French, Chinese and Korean learners of Japanese and native Japanese speakers (in Japanese)," Japanese-Language Education around the Globe, vol. 15, 2005, pp. 75–89.

[10] P. Tavakoli, "Pausing patterns: differences between L2 learners and native speakers," ELT Journal, vol. 65, no. 1, 2011, pp. 71–79.

[11] S. Okazawa, "Pauses and fillers in second language learners' speech," Studies in Language and Culture, vol. 23, 2014, pp. 52–66. [Online]. Available: http://opac.library.twcu.ac.jp/opac/repository/1/5697/ [retrieved: August, 2015]

[12] V. LoCastro, “Aizuchi: A Japanese conversational routine,” in Discourse Across Cultures: Strategies in World Englishes. Prentice Hall, 1987, pp. 101–113.

[13] A. Tera, K. Shirai, T. Yuizono, and K. Sugiyama, “Analysis of eye movements and linguistic boundaries in a text for the investigation of Japanese reading processes,” IEICE Transactions on Information and Systems, vol. E91.D, no. 11, 2008, pp. 2560–2567.

[14] S. K. Maynard, “On back-channel behavior in Japanese and English casual conversation,” Linguistics, vol. 24, no. 6, 1986, pp. 1079–1108.

[15] L. Miller, “Verbal listening behavior in conversations between Japanese and Americans,” in The Pragmatics of Intercultural and International Communication. John Benjamins Publishing, 1991, pp. 110–130.

[16] S. White, “Backchannels across cultures: A study of Americans and Japanese,” Language in Society, vol. 18, no. 1, 1989, pp. 59–76.

[17] T. Chen, “Existing research of Japanese backchannels : An overview for the future (in Japanese),” Japanese language education, vol. 2002, 2002, pp. 222–235.

[18] Twitter, Inc. The Streaming APIs. [Online]. Available: https://dev.twitter.com/streaming/overview [accessed: 2016-6-1]

[19] T. Moriyama, Y. Nitta, and H. Kudo, Japanese Grammar 3: Modality (in Japanese). Tokyo, Japan: Iwanami, 2000.

[20] H. Sawada, Modality (in Japanese). Tokyo, Japan: Kaitakusha, 2006.

[21] T. Masuoka, The Grammar of Modality (in Japanese). Tokyo, Japan: Kuroshio, 1991.

[22] R. E. Nisbett, The geography of thought : how Asians and Westerners think differently ... and why. New York: Free Press, 2003.

www.iariajournals.org

International Journal On Advances in Intelligent Systems
issn: 1942-2679

International Journal On Advances in Internet Technology
issn: 1942-2652

International Journal On Advances in Life Sciences
issn: 1942-2660

International Journal On Advances in Networks and Services
issn: 1942-2644

International Journal On Advances in Security
issn: 1942-2636

International Journal On Advances in Software
issn: 1942-2628

International Journal On Advances in Systems and Measurements
issn: 1942-261x

International Journal On Advances in Telecommunications
issn: 1942-2601