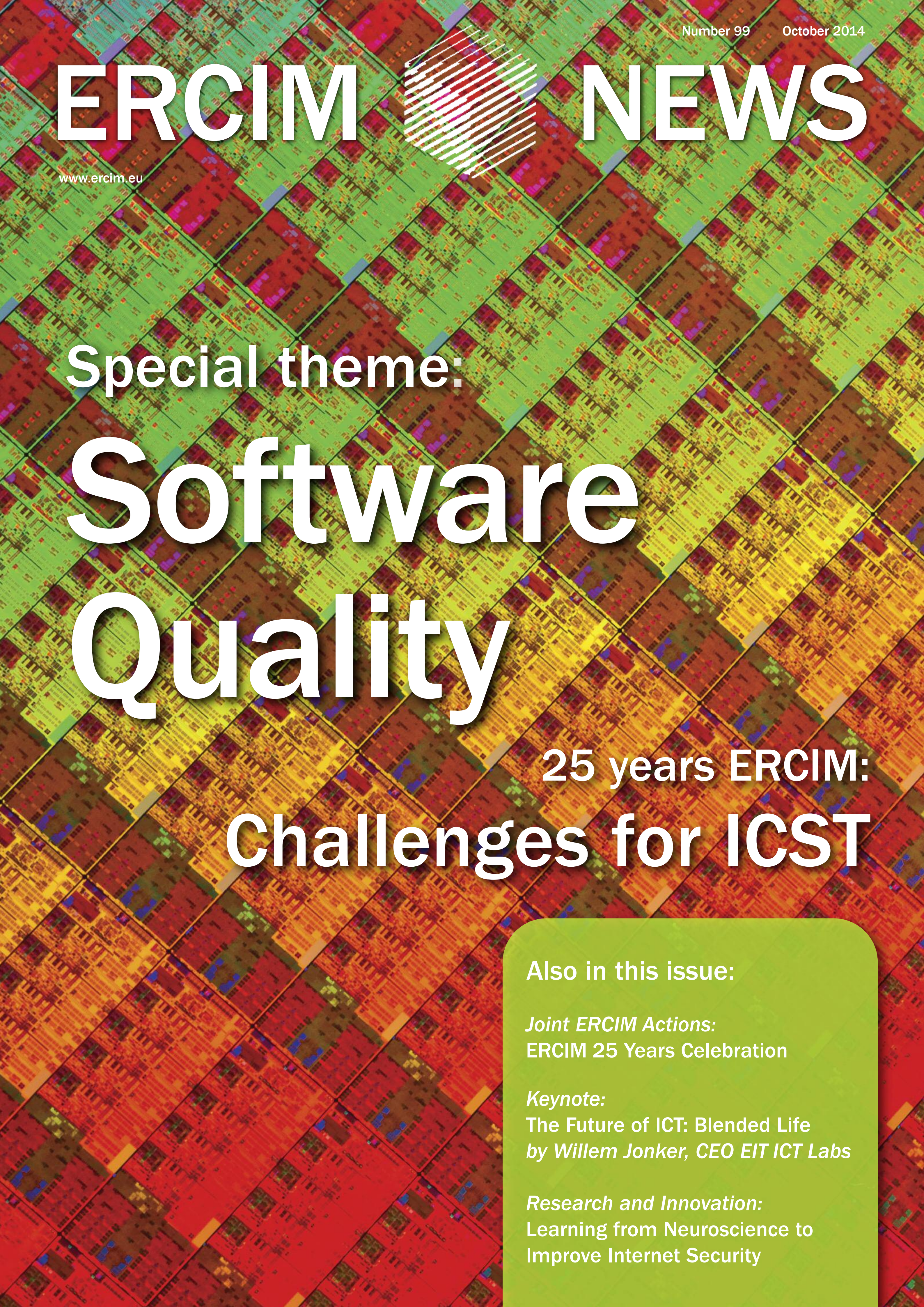
Number 99 October 2014

ERCIM NEWS

www.ercim.eu



Special theme:

Software Quality

25 years ERCIM: Challenges for ICST

Also in this issue:

Joint ERCIM Actions:
ERCIM 25 Years Celebration

Keynote:
The Future of ICT: Blended Life
by Willem Jonker, CEO EIT ICT Labs

Research and Innovation:
Learning from Neuroscience to Improve Internet Security

ERCIM News is the magazine of ERCIM. Published quarterly, it reports on joint actions of the ERCIM partners, and aims to reflect the contribution made by ERCIM to the European Community in Information Technology and Applied Mathematics. Through short articles and news items, it provides a forum for the exchange of information between the institutes and also with the wider scientific community. This issue has a circulation of about 6,000 printed copies and is also available online.

ERCIM News is published by ERCIM EEIG
BP 93, F-06902 Sophia Antipolis Cedex, France
Tel: +33 4 9238 5010, E-mail: contact@ercim.eu
Director: Jérôme Chailloux
ISSN 0926-4981

Contributions
Contributions should be submitted to the local editor of your country

Copyright notice

Advertising
For current advertising rates and conditions, see http://ercim-news.ercim.eu/ or contact peter.kunz@ercim.eu

ERCIM News online edition
The online edition is published at http://ercim-news.ercim.eu/

Subscription
Subscribe to ERCIM News by sending an email to en-subscriptions@ercim.eu or by filling out the form at the ERCIM News website: http://ercim-news.ercim.eu/

Next issue
January 2015, Special theme: Scientific Data Sharing

Cover image: an Intel processor wafer.
Photo: Intel Corporation.

Contents

JOINT ERCIM ACTIONS

KEYNOTE

25 YEARS ERCIM: FUTURE CHALLENGES FOR ICST

SPECIAL THEME

The special theme section "Software Quality" has been coordinated by Jurgen Vinju, CWI and Anthony Cleve, University of Namur.

Introduction to the Special Theme

RESEARCH AND INNOVATION

This section features news about research activities and innovative developments from European research institutes

EVENTS, IN BRIEF

Announcements

In Brief

ERCIM “Alain Bensoussan” Fellowship Programme

ERCIM offers fellowships for PhD holders from all over the world.

Topics cover most disciplines in Computer Science, Information Technology, and Applied Mathematics.

Fellowships are of 12-month duration, spent in one ERCIM member institute. Fellowships are proposed according to the needs of the member institutes and the available funding.

Conditions

Applicants must:

- have obtained a PhD degree during the last 8 years (prior to the application deadline) or be in the last year of the thesis work with an outstanding academic record
- be fluent in English
- be discharged or get deferment from military service
- have completed the PhD before starting the grant.

In order to encourage mobility:

- a member institute will not be eligible to host a candidate of the same nationality.
- a candidate cannot be hosted by a member institute, if by the start of the fellowship, he or she has already been working for this institute (including phd or postdoc studies) for a total of 6 months or more, during the last 3 years.

Application deadlines

30 April and 30 September

More information and application form:
http://fellowship.ercim.eu/

ERCIM 25 Years Celebration

The 25th ERCIM anniversary and the ERCIM fall meetings will be held at the CNR Campus in Pisa on 23-24 October 2014.

On the occasion of ERCIM’s 25th anniversary, a special session and panel discussion will be held on Thursday 23 October in the afternoon in the auditorium of the CNR Campus. Speakers and representatives from research, industry, the European Commission, and the ERCIM community will present their views on research and future developments in information and communication science and technology:

Programme

14:00 - 16:45

- Welcome address by **Domenico Laforenza**, President of ERCIM AISBL (CNR)
- **Alberto Sangiovanni Vincentelli**, Professor, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley: “Let’s get physical: marrying computing with the physical world”
- **Carlo Ratti**, Director, Senseable Lab, MIT, USA: “The Senseable City”
- **Alain Bensoussan**, International Center for Decision and Risk Analysis, School of Management, The University of Texas at Dallas, ERCIM co-founder: “Big data and big expectations: Is a successful matching possible?”
- **Rigo Wenning**, W3C’s legal counsel, technical coordinator of the EU STREWS project on Web security: presentation of the ERCIM White Paper “Security and Privacy Research Trends and Challenges”
- **Fosca Giannotti**, senior researcher at ISTI-CNR, head of the KDDLab: presentation of the ERCIM White Paper “Big Data Analytics: Towards a European Research Agenda”
- **Radu Mateescu**, senior researcher at Inria Grenoble - Rhône-Alpes, head of the CONVECS team: “Two Decades of Formal Methods for Industrial Critical Systems”
- **Emanuele Salerno**, senior researcher at ISTI-CNR: “MUSCLE: from sensing to understanding”

16:45 - 17:15 Coffee break

17:15 - 18:45
Panel: “ICT Research in Europe: How to reinforce the cooperation between the main actors and stakeholders”.
Panelists:

- **Carlo Ghezzi** (Moderator), President of Informatics Europe (Politecnico di Milano, Italy)
- **Domenico Laforenza**, President of ERCIM AISBL (CNR)
- **Fabio Pianesi**, Research Director, ICT Labs, European Institute for Innovations and Technology
- **Fabrizio Gagliardi**, President of ACM Europe
- **Jean-Pierre Bourguignon**, President of the European Research Council (ERC)
- **Paola Inverardi**, ICT Italian Delegate, Rector of the University of L’Aquila, Italy
- **Thomas Skordas**, Head of the FET Flagships Unit, European Commission, DG Communications Networks, Content and Technology (DG CNECT)

For more information, please contact:
Adriana Lazzaroni, CNR, Italy
E-mail: adriana.lazzaroni@iit.cnr.it

The Future of ICT: Blended Life

Willem Jonker , CEO EIT ICT Labs

Today we live a blended life. This blended life is a direct consequence of the deep penetration of Information and Communication Technology (ICT) into almost every area of our society. ICT brings ubiquitous connectivity and information access that enables disruptive innovative solutions to address societal megatrends such as demographic changes, urbanization, increased mobility and scarcity of natural resources. This leads to a blended life in the sense that the physical and virtual worlds are merging into one where physical encounters with friends and family are seamlessly integrated with virtual encounters on social networks. A blended life in the sense that work and private life can be combined in a way that offers the flexibility to work at any time from any location. A blended life combining work and life-long education facilitated by distance learning platforms that offer us a personalized path to achieving our life and career goals. Industries experience a blended life owing to the deep embedding of ICT into their production methods, products and services. Customers experience a blended life where ICT allows industries to include consumers in production, blending them into ‘prosumers’.

Blended life is becoming a reality and as such it brings both opportunities and challenges. On the one hand, it allows us to maintain better contact with people we care about, yet at the same time it is accompanied by a level of transparency that raises privacy concerns. The blending of private life and work has the clear advantage of combining private and professional obligations, but at the same time introduces the challenge of maintaining a work-life balance. The blending of products and services leads to personalization of offerings but, at the same time, the huge range of choice can be confusing for consumers. Blended production leads to shorter supply chains and cost-effective production, yet disrupts existing business models, resulting in considerable social impact.

Key drivers in the development of ICT itself, include future network technology (such as 5G, Internet of Things, Sensor Networks) at the communication layer and Cloud Computing at the information-processing layer (such as Software as a Service and Big Data Processing and Analytics). The main challenge here is to deal with the huge amounts of heterogeneous data both from a communication as well as an information processing perspective.

When it comes to the application of ICT in various domains, we see huge disruptions occurring both now and in the future in domains such as social networks, healthcare, energy, production, urban life, and mobility. Here the main challenge is to find a blending that simultaneously drives economic growth and quality of life. There are many domain-specific technical challenges, such as sensor technology for continuous health monitoring, cyber-physical systems for the industrial Internet, 3D-printing, smart-grids for energy supply, tracking and tracing solutions for mobility. Social, economic, and legal challenges are key to successful innovation in this area.

The issue of privacy is a prime example. The domains mentioned above are highly sensitive in regard to privacy. ICT that allows instant proliferation of information and continuous monitoring of behaviour can be perceived as a personal infringement; as a result several innovations have been slowed down, blocked or even reversed.

Innovations addressing societal challenges should involve social, economic, technical and legal specialists right from the inception, to map out the potential issues in a multidisciplinary way in order to ensure a proper embedding into society thus preventing these issues to become innovation blockers.

This approach is at the heart of EIT ICT Labs (www.eitictlabs.eu), a leading European player in ICT Innovation and Education supported by the European Institute of Innovation and Technology. Its mission is to foster entrepreneurial talent and innovative technology for economic growth and quality of life. EIT ICT Labs is active in many of the core ICT developments as well as the embedding of ICT in the above-mentioned domains. Education is an integral part of the EIT ICT Labs approach, since human capital is considered essential in bringing ICT Innovations to life.

Today we live a blended life. At the same time this blended life is only just beginning. Rapid developments in ICT will further drive the penetration of ICT into almost all areas of society leading to many disruptions. The key challenge ahead is to make sure that this blended life combines economic growth with high quality of life, which can only be achieved via a multidisciplinary innovation approach.

Willem Jonker

Future Challenges for ICST

Information and Communication Science and Technology (ICST) and Applied Mathematics are playing an increasing role helping to find innovative solutions raised by today's economic, societal and scientific challenges. ERCIM member institutes are at the forefront of European research in ICST and Applied Mathematics. Over the years ERCIM News has witnessed the significant advances made by scientists in this field and in the related application areas. On the occasion of ERCIM's 25th anniversary, we take a glance into the future with the following selection of articles, providing insight into just some of the multitude of challenges that our field is facing:

Economic/societal
- Intermediation Platforms, an Economic Revolution
  by Stéphane Grumbach
- Enabling Future Smart Energy Systems
  by Stefan Dulman and Eric Pauwels
- Will the IT Revolution Cost Our Children Their Jobs?
  by Harry Rudin

Science
- The Next Boom of Big Data in Biology: Multicellular Datasets
  by Roeland M.H. Merks

Software
- Looking Towards a Future where Software is Controlled by the Public (and not the other way round)
  by Magiel Bruntink and Jurgen Vinju
- Scaling Future Software: The Manycore Challenge
  by Frank S. de Boer et al.

Intermediation Platforms, an Economic Revolution

by Stéphane Grumbach

Intermediation platforms connect people, services and even things in ways that have been unthinkable until now. Search engines provide relevant references for people searching for information. Social networks connect users in their environment. Car pooling systems link drivers and passengers using the same routes. Intermediation platforms use big data to fuel the services they offer and these services are evolving extremely quickly but almost unnoticed. They are already competing with the oil industry as the world's top market capitalisations and are on the verge of revolutionising the world in which we live.

Intermediation platforms can connect people, services or things that share common or complementary interests and would benefit from knowing one another. It can be either users that seek out such connections or the platform itself which takes the initiative to suggest connections. The relevance of the intermediation, which relies on sophisticated algorithms and agile business models, ensures the success of the services provided by the platforms [1].

The first intermediation platforms that were deployed at a very large scale were search engines. Introduced in the late 1990s, the primary purpose of search engines is to connect people with the information they are looking for. Meanwhile their business model relies on their secondary service which is effectively targeting ads to users. Search engines rely on very complex algorithms to rank the data and provide relevant answers to queries. In the last decade, intermediation platforms have successfully penetrated an increasing number of sectors, mostly in the social arena.

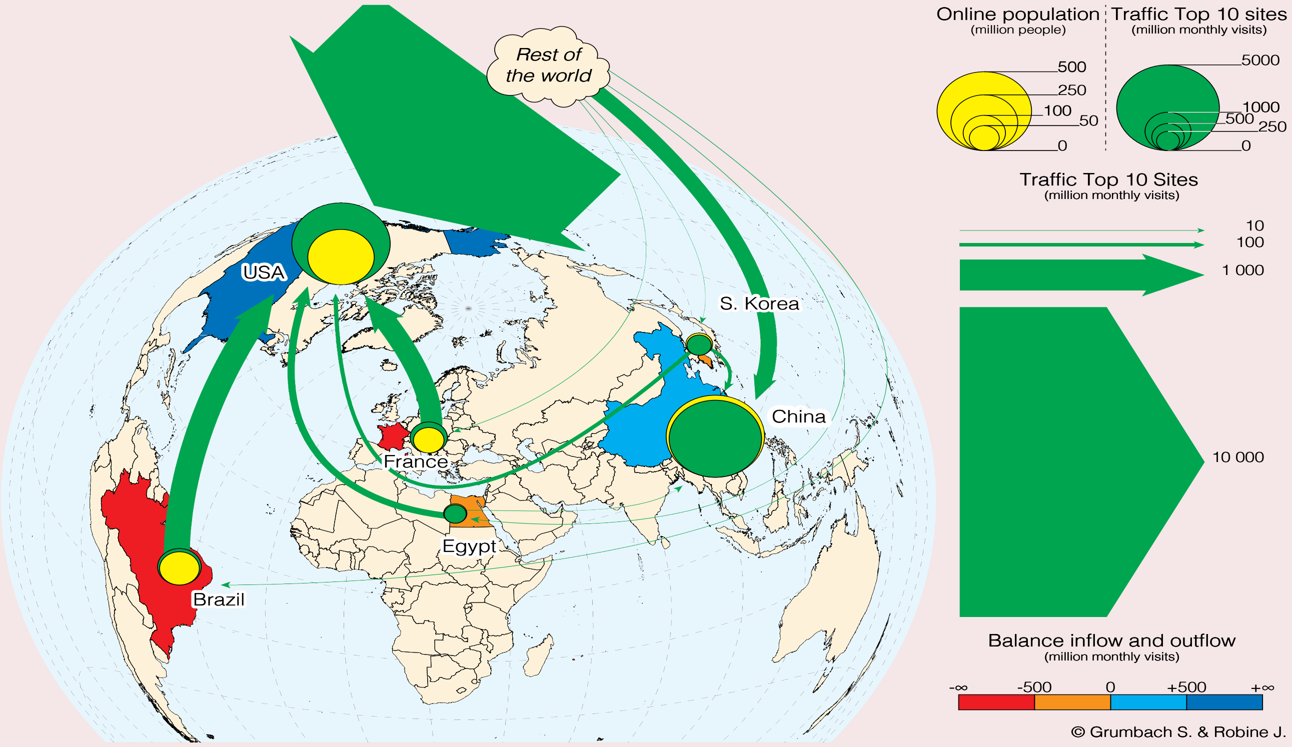
All intermediation platforms essentially rely on the same architecture. To begin with, they collect huge amounts of data which can come from the outside world (e.g., web pages for search engines) or be hosted by the platform (e.g., social networks). However, they are never produced by the platform itself but rather, by the people, services or things around it. These primary data are then indexed and transformed to extract information that fuels the primary services offered.

The activities of users on platforms generate secondary data. This secondary data essentially consists of traces which the platform generally has exclusive rights to, and allow the platform to create secondary services. A key example of this is the precise profiling of users which permits personalised and customised services: personal assistants trace users as they go about their day-to-day activities, not only online but also in the physical world through the use of geo-localization or quantified-self means.

Beyond personal services, platforms also generate data that can be of tremendous importance for virtually anything, for example, the information provided by search trends on search engines. Interestingly, it is hard to predict what uses this data might have and surprising applications of big data are emerging in many sectors. For instance, in the US, road traffic data from Inrix could reveal economic fluctuations before government services, much like Google Flu was ahead of the Centre for Disease Control. The externalities of big data, both positive and negative, need to be thoroughly considered and this is the goal of the European project BYTE [2], launched this year.

Platforms create ecosystems in which both users and economic players take a role. Platforms follow two rules: (i) they perform a gatekeeper role, acting as intermediaries for other services their users require and removing the need for other middleman; and (ii) they facilitate the easy development of services on their API for economic players. These two rules are fundamental to ensuring their capacity to collect data: this data then fuels all their services.

Platforms now have important economic power that rivals energy corporations. They offer services which have become essential utilities and are indispensable, not only for the general public but corporations as well. This latter group have come to rely on the services of platforms to facilitate customer relations and other fundamental aspects of their businesses. Like other essential utilities, such as water, energy and telecommunications, platforms provide service continuity, are non-discriminatory and can adapt to changes. Their business model is two-sided, with users on one side who



© Grumbach S. & Robine J.

Figure 1: An estimate of data flows between representative countries (as indicated by the arrows) that have been harvested online from the top ten websites in each country of origin. These websites represent about a third of the total activity of the entire top 500 sites. It is interesting to note that the US is clearly harvesting most of the data generated by most countries, including those in Europe.

receive free access to services and on the other side, clients (who at this stage mostly consist of advertisers) that can access special services offered to them.

In addition to offering brand new services, platforms are also disrupting [3] existing services by changing the rules of the game. By giving users easy access to a wide choice of possible services, platforms empower consumers. At the same time, they weaken traditional service providers. For instance, in the press, platforms allow users to access content from multiple sources. Such integrators might rely completely on algorithms and thus, bypass in-house journalists. In addition, by taking into account their reading habits or declared interests, platforms can offer a customised experience for their readers. This revolution is progressively taking place in all aspects of the content industry, including publishing, mass media, etc. This level of influence is not unique: there is hardly a sector that involves people, services or things that isn’t or wont be affected by platforms. Platforms are abolishing the distinction between service providers and consumers. Currently, the transportation and the lodging sectors are experiencing seriously impacts. This will soon be the case of media channels. States often react by trying to protect the suffering economic sectors.

Platforms empower people, but weaken establishments. By taking control of an increasing number of services, they also weaken States by disrupting the traditional economy and generating revenue for both users and the platform itself that escape taxation. To respond to this, tax systems must be reinvented. However, platforms also hold fantastic potential for meeting one of society’s most important challenges, the more frugal use of resources.

Platforms are on the verge of redesign the world as we know it. They not only introduce incredible promise, but also considerable challenges. To date, no platforms have been developed in Europe and consequently, its dependency on US platforms increases every day. This means that much of the European data harvested online is flowing to the US and with it an increasing loss of business opportunities and control over local activities.

Link:
BYTE project: http://byte-project.eu

References:

[1] S. P.Choudary: “Platform Thinking, The New Rules of Business in a Networked World”, http://platformed.info/
[2] S. P. Choudary, G. Parker and M. Van Alstyne: “Outlook 2014: Platforms are Eating the World”, http://www.wired.com/2013/12/outlook-2014-platforms-eating-world/

Please contact:

Stéphane Grumbach
Inria, France
E-mail: stephane.grumbach@inria.fr
https://who.rocq.inria.fr/Stephane.Grumbach
@SGrumbach

Enabling Future Smart Energy Systems

by Stefan Dulman and Eric Pauwels

The on-going transition to more sustainable energy production methods means that we are moving away from a monolithic, centrally controlled model to one in which both production and consumption are progressively decentralised and localised. This in turn gives rise to complex interacting networks. ICT and mathematics will be instrumental in making these networks more efficient and resilient. This article highlights two research areas that we expect will play an important role in these developments.

The confluence of various scientific and technological developments in computing, telecommunications and micro-electronics is rapidly ushering in an era in which humans, services, sensors, vehicles, robots and various other devices are all linked and interacting through loosely connected networks. Such networks can be thought of as cyber-physical systems as they build on an interwoven combination of the physical and digital environments, interacting through exchanges of data and control. Although the precise details of such cyber-physical systems will largely depend on the concrete application domain (e.g., logistics, smart energy grids, high precision agriculture), they tend to share a set of common important architectural characteristics. These include a large number of fairly autonomous components interacting at different levels, the routine collection and analysis of massive amounts of data, organic growth and fluidity in participation and an emphasis on decentralised decision making. This latter characteristic results in various levels of self-organisation or in other words a system of systems. The scale at which some of these systems are intended to operate, as well as their impact on society, calls for a principled approach to their design, analysis and validation. Therefore, these developments prompt challenging new research questions, creating novel directions of investigation and shedding new light on established ones.

Smart energy systems (SES) are an important case in point. The on-going transition to more sustainable energy production methods means that we are moving away from a monolithic, centrally controlled model to one in which both production and consumption are progressively decentralised and localised. Furthermore, the growing reliance on renewable energy sources such as wind and solar energy introduces considerable fluctuations to energy production that require a prompt and adequate response. As a result, energy networks are increasingly being twinned with parallel ICT networks that shuttle data, often in real time, between the various stakeholders (e.g., producers, distributors and consumers), with the ultimate goal of making the networks more efficient and resilient. Examples of this type of development include: the massive deployment of sensors to closely monitor network performance (e.g., the roll out of smart meters in many countries); the planned introduction of automated, online auctions and markets to use variable pricing as a stabilising mechanism for counter-acting fluctuations in consumption; and the ambition to decentralise energy production by parcelling the network into islands that are largely self-sufficient.

Source: Shutterstock

In a future of increasingly adaptive and opportunistically connected energy networks, control and self-organization based on the decentralized computation of global network parameters, will gain importance.

Obviously, in this short contribution it is impossible to highlight all the important research questions that are currently being explored to tackle the many challenges just mentioned. Rather, we will briefly outline two examples of such developments that piqued our personal interest.

A mathematical research topic that is currently being re-invigorated and has direct bearings on the design of SES is the use of gossip algorithms in distributed computing. As the name suggests, gossip algorithms attempt to compute aggregate (i.e., global) values for important network parameters by relying exclusively on local exchanges of information. Put differently, network nodes only talk to their neighbours, but nevertheless manage to efficiently compute reliable values for global network parameters, such as network-wide averages or maximum values. Such an approach obviates the need to establish a central control authority: since the resulting estimates diffuse across the network, it suffices to query an arbitrary node. Furthermore, these algorithms can be extended to be self-stabilising in the sense that changes in the network topology resulting from re-arrangements of neighbours are rapidly and automatically reflected in the final result.

From the short description above, it transpires that gossip-based aggregate computation addresses several of the key issues in SES. A distributed approach solves many scaling issues and proves to be robust against changes in network topology (e.g., nodes becoming unavailable due to failure or exit, new nodes joining the system). Furthermore, as the pro-

tocols rely mainly on anonymous data exchange, privacy issues are alleviated. Last but not least, significant progress is being made in the development of mathematical methodologies to precisely quantify the performance characteristics of these algorithms, in particular, error margins and speed of convergence (e.g., [1]).

A second challenge that is expected to drive future research finds its origin in the heterogeneity of the many ICT resources that are being combined in a typical SES setting. Indeed, although huge amounts of data about various aspects (e.g., production, consumption, storage, pricing, etc.) are routinely being collected, high quality information is still notoriously difficult to come by. Data are often siloed or served up in an unstructured and undocumented format. Data (and resources in general) need to come augmented with widely understood metadata in order to provide sufficient context that other agents can independent process it. This is of particular interest in the nascent field of cross-modal data mining which investigates methodologies that can be used to automatically combine information drawn from heterogeneous sources, for example, how can the content of messages on social media be linked to spikes in consumption data emanating from smart meters? To make this type of analysis possible it is imperative to develop more powerful semantic mediation tools that can automatically identify related concepts in different data sets and establish precise maps between them [2]. We expect that this will spur on further developments in semantic web technologies, in particular ontology alignment and RDF-reasoners.

Again, it is important to re-iterate that such a short contribution cannot do justice to all the different scientific disciplines that are expected to hugely impact SES, such as multi-scale and multi-physics simulations of networks, market mechanisms using multi-agent systems, privacy protection and cyber-security, just to name a few. However, we think that both the topics discussed here will feature prominently in future research and have an important impact, not just on SES research but more generally on the wider class of cyber-physical systems.

Link:
http://www.cwi.nl/research-themes/energy

References:

[1] D. Shah: Gossip Algorithms. Foundations and Trends in Networking, Vol. 3 (1), 2008,pp1-125.
[2] Choi, N., Song, I. Y., Han, H.: A survey on ontology mapping. ACM Sigmod Record, 35(3), 34-41 (2006)

Please contact:

Eric Pauwels
CWI, The Netherlands
E-mail: Eric.Pauwels@cwi.nl

Will the IT Revolution Cost Our Children Their Jobs?

by Harry Rudin

Over the past two and a half centuries, there have been several technological revolutions: the industrial revolution in around 1770, the introduction of steam engines and railways in the 1830s, the introduction of steel and electricity in the 1870s, the use of mass production from 1910 onwards and finally, the current Information Technology (IT) revolution that began in the 1970s. This modern day revolution is unique: not only has it brought us incredible achievements but it also poses some real threats to our traditional concept of employment.

Our Modern Day Revolution

In the past, technological revolutions have been followed by a stagnation in employment figures as traditional jobs are displaced by the new technologies. Then, as people adapt to those technologies and embrace their benefits, new jobs are created, productivity increases and overall living standards improve. This delay between technological breakthroughs and employment recovery is typically several decades [1]. In the case of the IT revolution, however, recovery seems to be taking much longer.

This prompts us to ask the question “Why?”. Previous technological introductions have also acted as a human labour multiplier. For example, the mechanical loom increased productivity by a factor of approximately 100 but is tiny in comparison with the progress seen in electronics and computers, as described by Moore’s Law. To illustrate this point, consider the invention of the transistor in around 1960: since the introduction of this device, computation costs have been reduced by a factor of 10^{10}. If a similar reduction was seen in aviation costs, it would equate to a brand new Airbus 320 costing one cent as opposed to 100 million dollars.

The IT revolution has been incredibly profound and therefore, it is not surprising that it has had profound consequences. IT goods and services have become relatively inexpensive and virtually ubiquitous and they have been applied to almost every aspect of our day-to-day lives to automate processes from mass production to routine bookkeeping. Now, they are poised to enhance, and in some instances even replace, intellectual processes that have long been thought to be the domain of human intellect alone. Today, IT is being used to guide robots, prepare legal dossiers and even play and win games that call for wit and context such as the popular US game show Jeopardy!.

IT, a Threat to Jobs?

While automation displaced blue-collar workers from repetitive, menial tasks, IT and robotics have accelerated and extended this trend into white-collar sectors. Further, the infinite capabilities of these technologies means they are rapidly making inroads into automating intellectual processes. In the future, it is highly likely that IT will drive our cars, fly

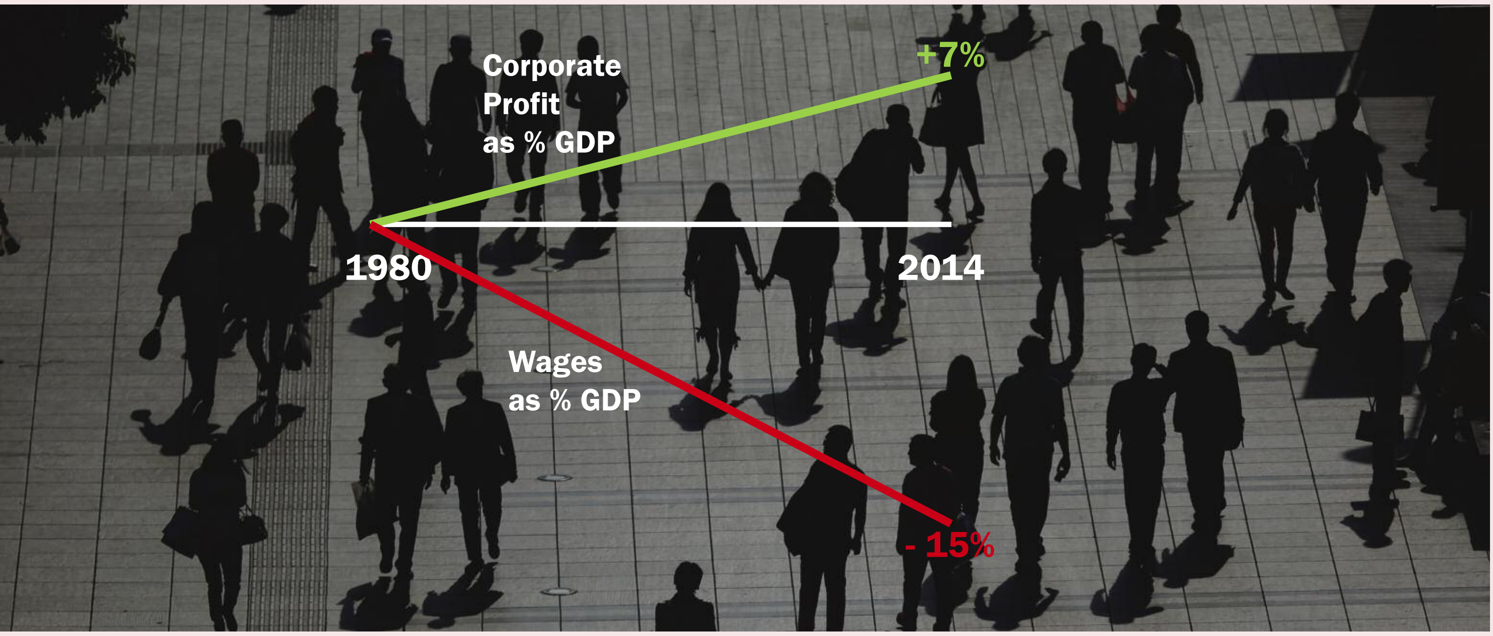


Figure 1: Corporate profits and wages as a percentage of gross domestic product in the US.

our planes, diagnose our diseases and manage our medical treatment. The term “white-collar” was used to described workers whose tasks required no manual dexterity. Now it refers to professionals who engage in unstructured and intellectual work, precisely the tasks that are increasingly being automated. In the last 35 years, corporate profits have grown, thanks largely to the increasing use of IT (Figure 1). Higher corporate profits mean firms have more available capital to invest expanding the IT-guided automation of their processes. As a result, many jobs have already become obsolete and this trend seems likely to continue. Simply put, the IT revolution has made machines cheaper than manpower and consequently, jobs are being lost and personal incomes are decreasing.

For the United States, Frey and Osborne [2] have predicted how jobs will develop over the next two decades. To do this they analysed more than 700 job types and arrived at some startling results. For instance, they predict (with a 90% probability) that telemarketers, bank tellers, insurance agents, file clerks and cashiers will all be replaced by IT. Bus drivers, teachers and flight attendants will also be heavily impacted, with half of all jobs predicted to be supplanted by IT. Overall, they predict an estimated 47% of all current jobs in the US will be eliminated. Psychologists, scientists, engineers and managers, however, appear to be some of the safer professions. Similar changes are taking place in here in Europe although luckily, the European education system better prepares students to deal with IT. More negatively, however, Europe is already facing overall unemployment rates of nearly 12%.

These employment changes are already well underway and this can be seen in the sharply decreasing rates of not only blue-collar incomes, but also some white-collar workers who perform well-structured tasks. In contrast, professionals with intellectual functions have enjoyed steady or increasing salaries. It is undeniable that this discrepancy is exacerbating the already unhealthy redistribution of wealth throughout the world (see C. Murray [3])

Our Challenge

The tidal wave of IT progress is a formidable force. In line with historical patterns, some economists are of the view that in time, society will learn how to embrace the new technologies, leverage IT to find new employment solutions and regain more of an equilibrium. However, the world has never dealt with a revolution that has such a profound scope. Happily, certain jobs will obviously be created. For example, there will be a growing demand for jobs that require a flair for creativity, personal interaction and social skills. In addition, huge numbers of software experts will be needed to adapt computers to handle tasks that are yet to be automated. Programming and analytical skills will also be in high demand. It is certainly true that IT will generate many new jobs, but will these replace the current jobs lost? Education will continue to play a vital role in meeting these future challenges: the financial advantages of having a good education are already evident.

Nonetheless we must do more. We have created some incredible technology and now the question is, can we now channel our inventiveness and ingenuity into creating new classes of work? New jobs that will result in a novel economy that is not simply based on consumable goods? The clock is running and I wish all of us, especially our children, good luck.

Link: http://www.zurich.ibm.com/~hr/IT_refs

References:

[1] C. Perez, “Technological Revolutions and Financial Capital: the Dynamics of Bubbles and Golden Ages,” Edward Elgar Publishing, 2002.
[2] C. B. Frey and M. A. Osborne, “The Future of Employment: How Susceptible Are Jobs to Computerisation,” Oxford University, 17 September, 2013.
[3] C. Murray, “Coming Apart: 1960-2010,” Random House, 2012.

Please contact:

Harry Rudin, Computer Networks
E-mail: hrudin@smile.ch

The Next Boom of Big Data in Biology: Multicellular Datasets

by Roeland M.H. Merks

Big data research in Life Sciences typically focuses on big molecular datasets of protein structures, DNA sequences, gene expression, proteomics and metabolomics. Now, however, new developments in three-dimensional imaging and microscopy have started to deliver big datasets of cell behaviours during embryonic development including cell trajectories and shapes and patterns of gene activity from every position in the embryo. This surge of multicellular and multi-scale biological data poses exciting new challenges for the application of ICT and applied mathematics in this field.

In 1995, when I was in the early stages of my biology masters and starting to think about PhD opportunities, Nature published a short feature article entitled 'The Boom in Bioinformatics'. The world needed bioinformaticians, the article said, "to take full advantage of the vast wealth of genetic data emanating from [...] the Human Genome Project and other [...] efforts." With a combined interest in computer science and biology, you might have thought therefore, this promised a bright future for me. The problem was bioinformatics did not excite me. I didn't believe it would solve the problem I had held closest to my heart since I had first seen tadpoles develop from eggs, embryogenesis. After all, embryos are not just bags of genes.

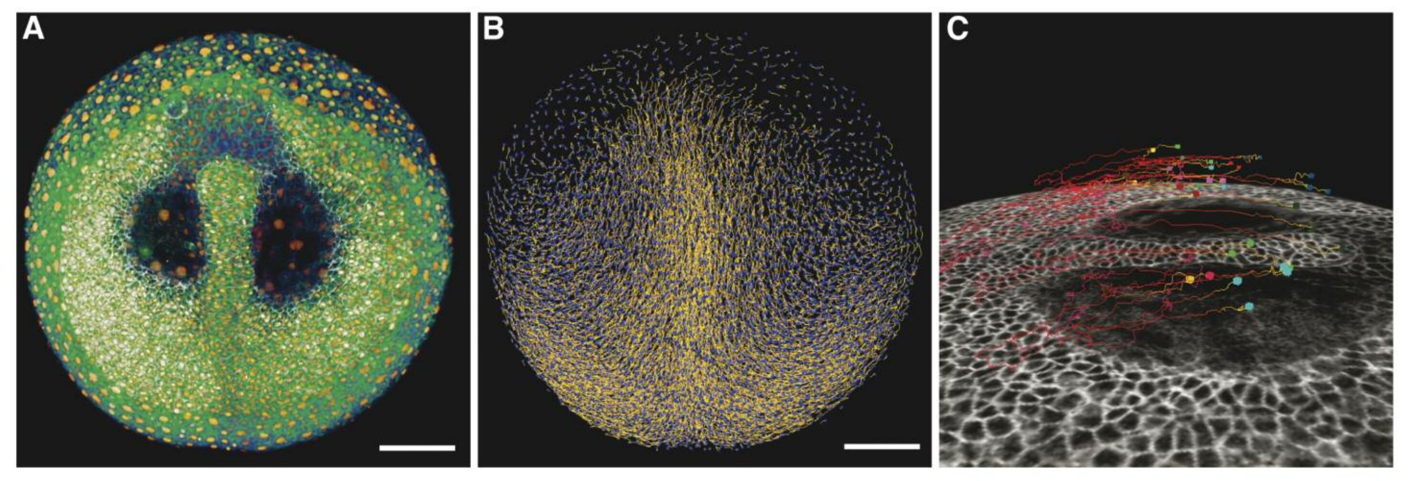
Technological developments in microscopy and image analysis are now producing a flood of new data that excites me much more. With this data, it is now possible to track the movements and behaviours of any cell, in an early embryo, organ, or tumour. With this capability we will now be able to identify what makes cells take a wrong turn in children with birth defects or how tumour cells can change their metabolism and movement to out compete their well-behaved neighbours and disrupt the structure and function of an organ. Such mechanistic insights will eventually make it possible to interfere with developmental mechanisms with a greater specificity than currently possible.

Conventional light microscopy can already follow the migration of a subset of individual cells (labelled with fluorescent markers) in organs but techniques are getting better. Two-photon microscopy techniques, used in conjunction with advanced image analysis, allow researchers to routinely generate all-cell datasets of developing embryos or organs. Applying this approach the BioEmergences platform at CNRS (Gif-sur-Yvette, France) recently produced a gene expression atlas featuring cellular resolution of developing zebrafish [1]. Soon we will be able to follow every cell in developing organisms and tissues and concurrently identify what genes they are expressing and what metabolites they are producing.

These developments will generate enormous new datasets that will be much bigger than those that currently exist in molecular biology. So how can we store such data in a structured way so that researchers can use it meaningfully to further the field of embryogenesis? First, the data must be stored in standard formats that facilitate sharing and make it accessible for on-going use after publication. For molecular data standards exist to ensure the gene sequence and functional features are captured. For example, Gene Ontology is a structured vocabulary that allows researchers to assign a readable list of well-defined biological functions to a gene. Two computer-oriented, domain-specific languages, SBML and CellML, can be used to describe the dynamic interaction networks of genes, proteins and metabolites. These languages create files that are similar to PDF files and they can be interpreted by many different software applications.

Ongoing initiatives in the field of information sciences are laying the foundations for similar data standards and domain-specific languages in the multicellular biology community. New versions of SBML will allow users to describe the distribution of molecules in fixed geometries and coupled cells. However, in a recent paper that proposed a Cell Behaviour Ontology (CBO) [2], it was argued that SBML is not the most efficient or insightful way to annotate embryological data. The multicellular organism is a collection of thousands to trillions of individual cells. Individually describing the gene expression levels and biophysical properties of each cell will create huge datasets but not necessarily yield useful insights. Even the most detailed three-dimensional movies or sets of cell trajectories are merely pretty pictures unless we can identify and label their components meaningfully. A useful comparison is thinking about the difference between providing a list of pixels in an image versus the list of things in that image. CBO focuses on describing the behaviour of cells and the dependency of those behaviours on the cell's internal machinery. This includes its gene expression pattern and local environment. This declarative approach allows the CBO to categorise each cell in a developing embryo using a manageable set of cell types which range from the tens to hundreds in number. Each cell type is characterised by the same class of behaviours, thus, cells belonging to the same cell type share the same behaviours. Each cell follows a set of logical input and output rules that guide these behaviours and its transition from one cell type to another (i.e., differentiation). Many cell types in multicellular organisms are 'sub-types' whose behaviour varies in subtle ways around a general 'base' cell type. For example, the endothelial cells in a developing blood vessel are made up of two sub-types: 'tip' cells at the end of a sprouting blood vessel which are usually more spikey and motile and 'stalk' cells which occur to the back of the sprout. This approach allows the CBO to develop a hierarchical classification of cell types and cell behaviours.

Besides compressing the data, the classification of cell behaviours will also enable quantitative biologists to understand biological development to a point that, with the aid of applied mathematicians, they can then reconstruct it using agent-based computer simulations. This will then enable them to unravel how subtle changes in cell behaviour, driven by factors such as inherited disease or cancer, can affect the outcome of development and why. Thus, the resulting



datasets become more meaningful descriptions of the observations as well as sets of rules to construct agent-based computer simulations of those observations. In this way, CBO takes a 'cell-based approach' [3], which views embryogenesis as the collective behaviour of a 'colony' of individual cells.

The extraction of cell behaviours from data, followed by the re-synthesis of the embryo as a computer simulation is already under way. At Inria (Roquencourt, France), a team led by Dirk Drasdo is using structural images to build simulations of liver regeneration following poisoning. At Inria (Montpellier, France), the VirtualPlants team headed by Christophe Godin has used detailed plant tissue images to build cell-level simulations of leaf initiation and vascular development in plants. In our own work here at CWI (Amsterdam), we are simulating the formation of blood vessel sprouts, e.g., during cancer neoangiogenesis, from the chemical and mechanical interactions between endothelial cells. As multicellular imaging datasets are merging with explanatory computer modelling, big data in biology is finally starting to really excite me.

Links:
Multicellular Modeling at CWI
http://biomodel.project.cwi.nl
BioEmergences project: http://bioemergences.iscpif.fr
VirtualPlants team: http://www.inria.fr/en/teams/virtual-plants
Dirk Drasdo group: http://ms.izbi.uni-leipzig.de
CompuCell3D multicellular simulator:
http://compucell3d.org

References:
[1] C. Castro-González et al.: "A Digital Framework to Build, Visualize and Analyze a Gene Expression Atlas with Cellular Resolution in Zebrafish Early Embryogenesis", PLoS Comp. Biol., 10 (6), 2014, e1003670
[2] J.P. Sluka, J. P. et al .: "The Cell Behavior Ontology: Describing the Intrinsic Biological Behaviors of Real and Model Cells Seen as Active Agents", Bioinformatics, 30(16), 2014, 2367-2374
[3] R.M.H. Merks and J.A. Glazier: "A Cell-Centered Approach to Developmental Biology", Physica A, 352(1), 2005, 113–130.

Figure: Zebrafish (Danio rerio) imaged live throughout early development (gastrulation). Snapshots of the tailbud stage.
A: Raw data (fluorescent nuclei and membranes), display with avizo software, data cut at a depth of 100 microns.
B: Display of detected nuclei and cell trajectories, calculated using the BioEmergences workflow (http://www.bioemergences.eu).
A, B scale bar 100 microns. Close up in C to show selected clones (colored cubes) and their trajectories for the past 6 hours, in white an orthoslice of the membrane channel.
Pictures by Nadine Peyriéras, BioEmergences.eu, CNRS Gif-sur-Yvette, France.

Please contact:
Roeland Merks, CWI, The Netherlands
E-mail: Roeland.Merks@cwi.nl

Looking Towards a Future where Software is Controlled by the Public (and not the other way round)

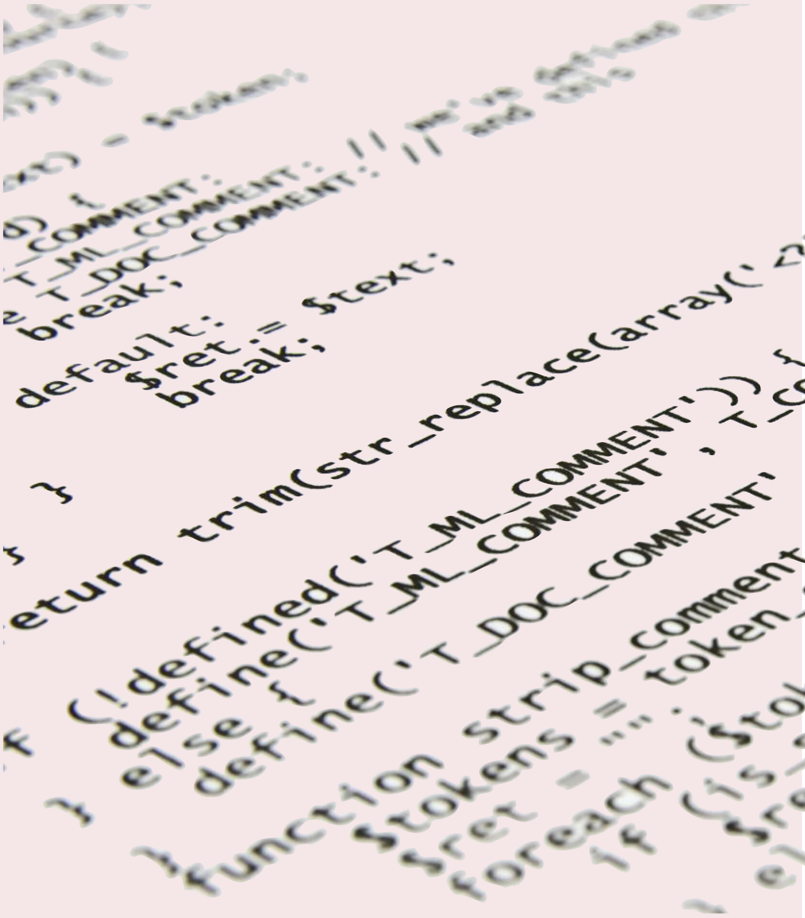
by Magiel Bruntink and Jurgen Vinju

Nowadays, software has a ubiquitous presence in everyday life and this phenomenon gives rise to a range of challenges that affect both individuals and society as a whole. In this article we argue that in the future, the domain of software should no longer belong to technical experts and system integrators alone. Instead it should transition to a firmly engaged public domain, similar to city planning, social welfare and security. The challenge that lies at the heart of this problem is the ability to understand, on a technical level, what all the different software actually is and what it does with our information.

Software is intrinsically linked to many of the challenges currently facing society. The most obvious of these challenges is data security and user privacy. Much of the software currently in use collects data. This data comes from a wide range of sources including recreational activities, personal health, messaging, street surveillance, financial transactions and international communications. Software is not only deployed on personal (mobile) computing devices but also through far-reaching government programs. In all cases, it is the software that tells each computing device how to participate in the act of data collection and process the activities that bring benefits. Whether these benefits are for the greater good of society, however, is not always clear cut. This prompts questions such as “Who is aware of the exact data collected by their smartphone, or where (on the internet) it will be stored?”, “What servers hold the contents of your software-supported tax return and in which countries are they located?” or “Is there a database somewhere that somehow stores a picture of you linked to a crime scene?”.

Besides the obvious political and social aspects of these questions, there are more fundamental problems that still need to be addressed by software researchers and practitioners. The core problem that exacerbates the issues of software security and privacy is that software is not sufficiently well understood at a technical level, especially at the scales at which it is now being developed and deployed. All too often, software is so complex it can’t even be handled by the most experienced software engineers or explained by the most advanced theories in computer science, and too big to be summarised accurately by the automated tools created for that purpose. How then are policy makers or the general public supposed to be able to make software-related decisions that are based on facts and insight?

Given that software complexity is still an untamed problem, what consequences exist for data security and privacy? There are numerous examples of incidents in which the security of key systems in many (public) organizations have been breached. Recently, a serious vulnerability dubbed the ‘Heartbleed bug’ was exposed in a software (OpenSSL) that is supposed to secure vast numbers of Internet servers. It isn’t at all clear what happened to the data stored on those systems that were compromised by these vulnerabilities.



Source: Shutterstock

```
) - $token;
_COMMENT:
T_ML_COMMENT: // we've defined this
T_DOC_COMMENT: // and this
break;
default:
    $ret .= $text;
    break;
}
return trim(str_replace(array('
if (!defined('T_ML_COMMENT')) {
    define('T_ML_COMMENT', T_C
} else {
    define('T_DOC_COMMENT'
}
function strip_comment
    $tokens = token_
    $ret = "";
    foreach ($tok
        if (is_
            $r
```

The core problem that exacerbates the issues of software security and privacy is that software is not well enough understood on a technical level, especially at the scale at which it is now being developed and deployed.

Heartbleed, in particular, provides an interesting illustration of the level of software complexity we are dealing with. The bug itself consists of only two lines of code, whereas the entire OpenSSL software package contains 450,000 lines of code [1]. Industry research into the existence of bugs or defects suggests a wide range in the bug ratio, from 0.1 to 100 bugs per 1,000 lines of code [2,3]. This ratio is strongly related to how well the software was developed and tested. Clearly, our current understanding of software does not allow us to develop software without bugs and is just one of the consequences of software complexity. Other considerations include the high cost and lack of performance.

Considering all this, we feel that the future of software should involve a radical change, best summarised as follows:

Software complexity should become a public problem, instead of simply remaining just a problem for the public. In our view, the current situation in which software is too complex to be handled properly should transition to a situation where software-related decisions can feasibly be made by non-experts, in particular policy makers and citizens.

In our view, a positive development has been the installation of the (temporary) committee on ICT by the Dutch House of Representatives, which is tasked with investigating several problematic e-government projects. We envision a similar public status for software as given to law making, city planning, social security, etc. While all these social priorities still require a certain level of technical expertise, public debate determines their direction. There is a long road ahead to reach a point where software can join this list. We feel the following directions are essential to making this journey:

- investment in research that creates more accessible software technologies, for instance, domain-specific (programming) languages that achieve a better fit to societal problems and reduce software complexity;
- investment in empirical research that considers the current state-of-the-art practices in dealing with software complexity with a view to scientifically establishing good and bad practices, methods and technologies;
- the introduction of software and computing curriculum at the primary, secondary and higher levels of education to increase general software literacy and ultimately, foster a better public understanding of software complexity; and
- contributions to the public debate on the nature of software and its impacts on society, for instance, by arguing that society-critical software should transition to open-source models, enabling public control and contribution.

In conclusion, to arrive at a future where software is something we can all understand and control, as opposed to us being controlled by software and its complexities, a strong focus on software will be required in both research and education. Therefore, it is high time to generate public engagement on the complexities of software and the challenges that it creates.

Link:

[1] Blackduck Open Hub:
https://www.openhub.net/p/openssl

References:

[2] Coverity Scan: 2013 Open Source Report, http://softwareintegrity.coverity.com/rs/coverity/images/2013-Coverity-Scan-Report.pdf
[3] Watts S. Humphrey, The Quality Attitude, SEI, 2004, http://www.sei.cmu.edu/library/abstracts/news-at-sei/wattsnew20043.cfm

Please contact:

Magiel Bruntink,
University of Amsterdam, The Netherlands
E-mail: M.Bruntink@uva.nl
or
Jurgen Vinju
CWI, The Netherlands
E-mail: Jurgen.Vinju@cwi.nl

Scaling Future Software: The Manycore Challenge

by Frank S. de Boer, Einar Broch Johnsen, Dave Clarke, Sophia Drossopoulou, Nobuko Yoshida and Tobias Wrigstad

Existing software cannot benefit from the revolutionary potential increases in computational power provided by manycore chips unless their design and code are polluted by an unprecedented amount of low-level, fine-grained concurrency detail. As a consequence, the advent of manycore chips threatens to make current main-stream programming approaches obsolete, and thereby, jeopardizes the benefits gained from the last 20 years of development in industrial software engineering. In this article we put forward an argument for a fundamental breakthrough in how parallelism and concurrency are integrated into the software of the future.

A radical paradigm shift is currently taking place in the computer industry: chip manufacturers are moving from single-processor chips to new architectures that utilise the same silicon real estate for a conglomerate of multiple independent processors known as multicores. It is predicted that this development will continue and in the near future multicores will become manycores. These chips will feature an estimated one million cores. How will this hardware development affect the software? The dominant programming paradigm in the industry today is object-orientation. The use of objects to structure data and drive processing operations in software programs has proven to be a powerful concept for mastering the increasing complexity of software. As a consequence, in the last few decades, industry has invested heavily in object-oriented software engineering practices.

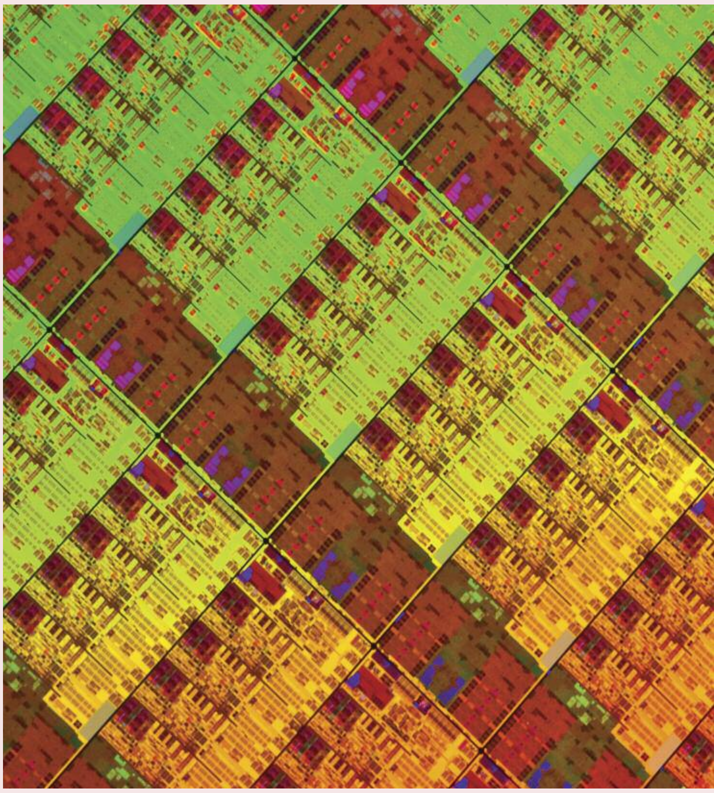
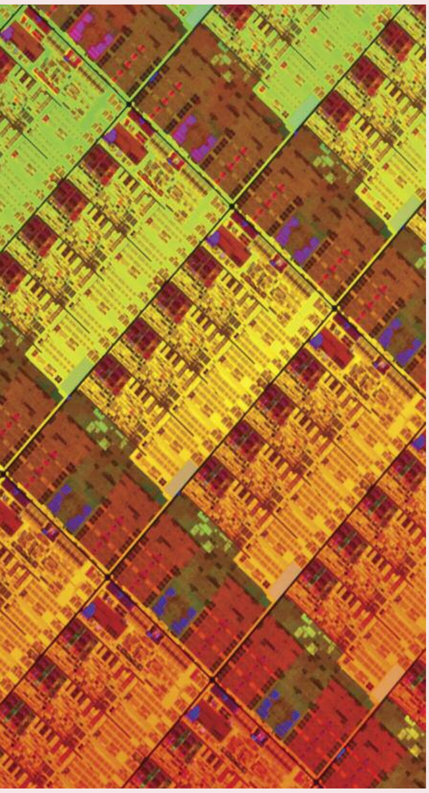


Photo: Intel Corporation

However, the current concurrency model commonly used for object-oriented programs in industry is multithreading. A thread is a sequential flow of control which processes data by invoking the operations of the objects storing the data. Multithreading is provided through small syntactic additions to the programming language which allow several such threads to run in parallel. Nevertheless, the development of efficient and precise concurrent programs for multicore processors is very demanding. Further, an inexperienced user may cause errors because different parallel threads can interfere with each other, simultaneously reading and writing the data of a single object and thus, undoing each other's work. To control such interference, programmers have to use low-level synchronization mechanisms, such as locks or fences, that feature subtle and intricate semantics but whose use is error-prone. These mechanisms can be introduced to avoid interference but generate additional overhead that is caused by threads that need to wait for one another, and thus, cannot be run in parallel. This overhead can also occur because the data are distributed across different parts of the architecture (i.e., cache and memory). If the data access pattern used by the various threads does not match their distribution pattern, the program generates a large amount of overhead transferring data across processors, caches and memory.

To address these issues, increasingly advanced language extensions, concurrency libraries and program analysis techniques are currently being developed to explicitly control thread concurrency and synchronization. However, despite these advances in programming support, concurrency is still a difficult task. Only the most capable programmers can explicitly control concurrency and efficiently make use of the relatively small number of cores readily available today.

Thus, manycore processors require radically new software abstractions to coordinate interactions among the concurrent processes and between the processing and storage units. This task requires a fundamental breakthrough in how parallelism and concurrency are integrated into programming languages, substantiated by a complete inversion of the current canonical language design. By inverting design decisions, which have largely evolved in a sequential setting, new programming models can be developed that are suitable for mainstream concurrent programming and deployment onto parallel hardware. This could be achieved without imposing a heavy syntactic overhead on the programmer.



A chip processor wafer. Chip manufacturers are moving from single-processor chips to new architectures that utilise the same silicon real estate for a conglomerate of multiple independent processors known as multicores.

The authors of this article are the principal investigators of the three year EU Upscale project (From Inherent Concurrency to Massive Parallelism through Type-Based Optimizations) which started in March 2014. In this project we take as starting point of the inverted language design existing actor-based languages [1] and libraries (i.e., Akka Actor API; see Links section). In contrast to an object, an actor executes its own thread of control in which the provided operations are processed as requested, by the actors which run in parallel. These requests are processed according to a particular scheduling policy, e.g., in order of their arrival. Sending a request to execute a provided operation involves the asynchronous passing of a corresponding message. That is, the actor that sends this message continues the execution of its own thread. Both concurrency and the features which typically make concurrency easier to exploit, such as immutability, locality and asynchronous message passing, will be default behaviour of the actors. This inversion produces a programming language that can be easily analysed as properties which may potentially inhibit parallelism (e.g., synchronous communication and shared mutable state) must be explicitly declared.

The key feature of the Encore language that is currently under development is that everything will be designed to leverage deployment issues. Deployment is the mapping of computations to processors and the scheduling of such computations. The main rationale of the inverted language design is to support the automated analysis of the code in order that deployment-related information can be obtained. This information can then be used to facilitate optimisations by both the compiler and at run-time. These automated optimisations will alleviate the design of parallel applications for manycore architectures and thus will make the potential computing power of this hardware available to mainsteam developers.

Links:

Upscale project: http://www.upscale-project.eu
Akka Actor API. http://akka.io/docs/

Reference:

[1] Gul A. Agha: "ACTORS - a model of concurrent computation in distributed systems", MIT Press series in artificial intelligence, MIT Press 1990, ISBN 978-0-262-01092-4, pp. I-IX, 1-144

Please contact:

Frank S. de Boer
CWI, The Netherlands
Tel: +31 20 5924139
E-mail: F.S.de.Boer@cwi.nl

Introduction to the Special Theme

Software Quality

by Jurgen Vinju and Anthony Cleve, guest editors for the special theme section

The introduction of fast and cheap computer and networking hardware enables the spread of software. Software, in a nutshell, represents an unprecedented ability to channel creativity and innovation. The joyful act of simply writing computer programs for existing ICT infrastructure can change the world. We are currently witnessing how our lives can change rapidly as a result, at every level of organization and society and in practically every aspect of the human condition: work, play, love and war.

The act of writing software does not imply an understanding of the resulting creation. We are surprised by failing software (due to bugs), the inability of rigid computer systems to "just do what we want", the loss of privacy and information security, and last but not least, the million euro software project failures that occur in the public sector. These surprises are generally not due to negligence or unethical behaviour but rather reflect our incomplete understanding of what we are creating. Our creations, at present, are all much too complex and this lack of understanding leads to a lack of control.

Just as it is easy to write a new recipe for a dish the world has never seen before, it is also easy to create a unique computer program which does something the world has never seen before. When reading a recipe, it isn't easy to predict how nice the dish will taste and, similarly, we cannot easily predict how a program will behave from reading its source code. The emergent properties of software occur on all levels of abstraction. Three examples illustrate this. A "while loop" can be written in a minute but it can take a person a week or even a lifetime to understand whether it will eventually terminate or not on any input. Now imagine planning the budget for a software project in which all loops should terminate quickly. Or take a scenario where you simply need to scale a computer system from a single database with a single front-end application to a shared database with two front-end applications running in parallel. Such an "improvement" can introduce the wildest, unpredictable behaviours such as random people not getting their goods delivered, or worse, the wrong limb amputated. In the third example, we do not know how the network will react to the load generated by the break of the next international soccer match between France and Germany, e.g., "When will it all crash?".

Higher quality software is simpler software, with more predictable properties. Without limiting the endless possibilities of software, we need to be able to know what we are creating. Teaching state-of-the-art software engineering theory and skills is one way of improving understanding but alone, this is not enough. We are working on developing better theories and better tools to improve our understanding of complex software and to better control its complex emergent behaviours. We will be able to adapt existing software to satisfy new requirements and to understand how costly these adaptations will be and the quality of the results. We will be able to design software in a way that means that consciously made design decisions will lead to predictable, high quality software artifacts. We will be able to plan and budget software projects within reasonable margins of error.

In this special theme of ERCIM News, some of the recent steps developed to understand and manipulate software quality are presented. We aren't yet at the stage where we fully understand, or can control software but we are certainly working towards this point. Some researchers are studying the current reality of software, discovering theories and tools that can improve our abilities to analyse, explain and manipulate. Other researchers are re-thinking and re-shaping the future of software by discovering new, simpler languages and tools to construct the next generation of software. These two perspectives should leapfrog us into a future where we understand it all.

As quality and simplicity are highly subjective concepts, our best bet is to strive to increasingly contextualising software engineering theory and technology. General theory, languages and tools have resulted in overly complex systems so now, more specialised tools and techniques for distinct groups of people and industries are being discovered. For example, instead of modelling computation in general, we are now modelling big data processing; instead of inventing new general purpose programming languages, we are now focusing on domain specific formalisms; and instead of reverse engineering all knowledge from source code, we are now extracting domain specific viewpoints.

We hope you will find this selection of articles an inspiring overview of state-of-the-art software quality engineering research and beyond.

Please contact:
Jurgen Vinju
CWI and TU Eindhoven, The Netherlands
E-mail: Jurgen.Vinju@cwi.nl

Anthony Cleve
University of Namur, Belgium
E-mail: anthony.cleve@unamur.be

Monitoring Software Quality at Large Scale

by Eric Bouwers, Per John and Joost Visser

In 2004, the Software Improvement Group (SIG) introduced a new software monitoring service. Then a recent spin-off of CWI with a couple of clients and a vision, SIG has transformed itself ten years later into a respected (and sometimes feared) specialist in software quality assessment that helps a wide variety of organizations to manage their application portfolios, their development projects, and their software suppliers.

Over the last 10 years, a range of technological, organizational and infrastructure innovations have allowed SIG to grow to the point that it provides an assessment service currently processing 27 million lines of code each week. In this article, we present a brief discussion of a few of those innovations.

Analysis tools

The software analysts at SIG are also software developers who create and continuously improve their own suite of software analysis tools. Not only are these tools adept at picking apart source code, they can be easily extended to support a range of additional computer languages. To date, this strength has been exploited to develop support for around 100 different languages, 50 of which are used on a continuous basis. These tools are also good at operating autonomously and scale appropriately. After an initial configuration, new batches of source code can be automatically analyzed quickly, allowing the analysts to focus their attention on the quality anomalies that are found [1]. On average, across all systems types, serious anomalies occur in approximately 15% of the analysed code.

Evaluation models

Whilst all software systems differ (i.e., their age, technologies used, functionality and architecture), common patterns do exist between them. These become apparent through on going, extensive analysis. SIG's analysts noticed these patterns in the software quality measurements and consolidated their experience to produce standardized evaluation models that operationalize various aspects of software quality (as defined by the ISO-25010 international standard of software product quality). The first model focused on the “maintainability” of a system. First published in 2007, this model has since been refined, validated and calibrated against SIG's growing data warehouse [2]. Since 2009, this model has been used by the Technischer Überwachungs-Verein (TÜV) to certify software products.

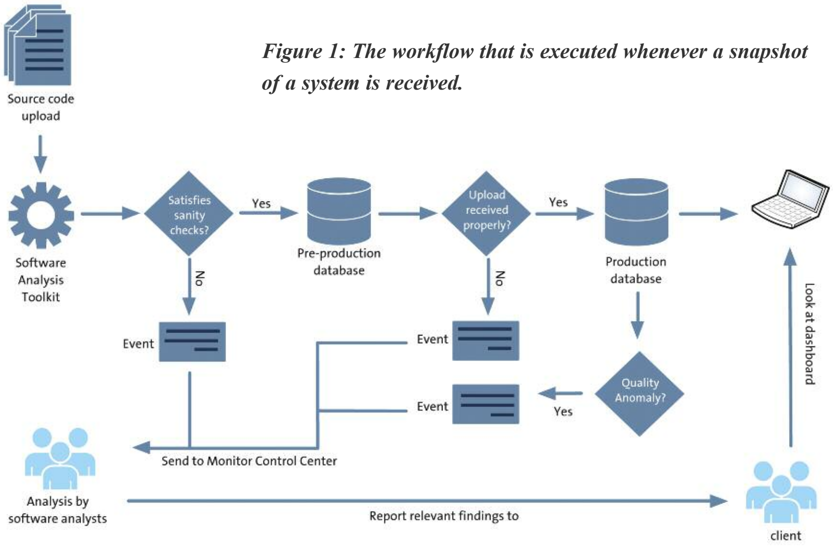


Figure 1: The workflow that is executed whenever a snapshot of a system is received.

Recently, two applications used by the Dutch Ministry of Infrastructure (Rijkswaterstaat) to assist with maintaining safe waterways, were awarded 4-star quality certificates by this organisation (from a possible 5 stars). Similar models for software security, reliability, performance, testing and energy-efficiency, have recently become available and these are continuously being refined.

Lab organization

Scalable tools and models that can effectively be applied are extremely valuable, but to achieve this, proper organization is paramount. SIG organizes its software analysis activities in an ISO-17025 certified lab. This means that analysts undergo proper training and follow standardized work procedures, consequently producing reliable measurement results that can be repeated. When quality anomalies are detected, they undergo triage in the Monitor Control Centre (Figure 1). Here, the false positive results are separated out. Then, depending on the severity and/or type of finding, the analyst works with the client to determine an appropriate resolution. If the anomaly cannot be resolved, then senior management becomes involved. Currently, SIG monitors over 500 software systems and takes over 200 code snapshots each week. From these, their analysts are responsible for assessing over 27 million lines of code, in over 50 different languages from COBOL and Scala to PL/SQL and Python.

Value adding: beyond software quality

On the foundation of tools and models, SIG has built an advisory practice. Working together with the analysts, the role of the advisors is to translate technical software quality findings into risks and recommendations. Thus, SIG is able to provide cost comparisons (e.g., the cost of repairing quality defects versus not repairing them [3]) or provide project recommendations (e.g., suboptimal quality may be a reason to postpone deployment, cancel a project or provide better conditions to the developers). By providing this context to the software findings, SIG offers meaningful value for its client's technical staff and decision-makers.

Ongoing R&D

The growth of SIG thus far, and its future path depends on ongoing investment in R&D. Into the future, SIG is looking to keep working with universities and research institutes from around the Netherlands (including Delft, Utrecht, Amsterdam, Tilburg, Nijmegen and Leiden) and beyond (e.g., the Fraunhofer Institute) to explore new techniques to control software quality. A number of questions still remain unanswered. For example, "how can the backlogs of agile projects be analysed to give executive sponsors confidence in what those agile teams are doing?", "how can security vulnerabilities due to dependencies on third-party libraries be detected?", "how can development teams be given insight into the energy footprint of their products and ways to reduce them?" or, "what are the quality aspects of the software-defined infrastructure that support continuous integration and deployment?". By continuing to explore the answers to these questions and others, SIG will continue to grow in the future.

References:

[1] D. Bijlsma, J. P. Correia, J. Visser: "Automatic event detection for software product quality monitoring", QUATIC 2012

[2] R. Baggen et al.: "Standardized Code Quality Benchmarking for Improving Software Maintainability", Software Quality Journal, 2011

[3] A. Nugroho, T. Kuipers, J. Visser: "An empirical model of technical debt and interest", MTD 2011.

Please contact:
Eric Bouwers, Per John or Joost Visser
Software Improvement Group,
The Netherlands
E-mail: e.bouwers@sig.eu, p.john@sig.eu or j.visser@sig.eu

OSSMETER: A Health Monitoring System for OSS Projects

by Nicholas Matragkas, James Williams and Dimitris Kolovos

OSSMETER is a FP7 European project that aims to extend the state-of-the-art in the field of automated analysis and the measurement of open-source software (OSS). It also aims to develop a platform that will support decision makers in the process of discovering, comparing, assessing and monitoring the health, quality, impact and activity of OSS.

Deciding if an open-source software (OSS) project meets the standards required for adoption, in terms of quality, maturity, the activity of the development and user support, is not a straightforward process. It involves exploring a variety of information sources including OSS source code repositories, communication channels (e.g., newsgroups, forums and mailing lists) and bug-tracking systems. Source code repositories help the potential adoptee to identify how actively the code has been developed, which programming languages were used, how well the code has been commented and how thoroughly the code has been tested. Communication channels can identify whether user questions are being answered in a timely and satisfactory manner and help estimate how many experts and users the software has. Finally, bug-tracking systems can show whether the software has many open bugs and the rate at which those bugs are fixed. Other relevant metadata such as the number of downloads, the license(s) under which the software is made available and its release history may also be available from the forge that hosts the project. If available, this myriad of information can help OSS adoptees make informed decisions, however, the disaggregated nature of the information makes this analysis tedious and time consuming.

This task becomes even more challenging if the user wishes to identify and compare several different OSS projects that offer software with similar functionality (e.g., there are more than 20 open source XML parsers for the Java programming language) and make an evidence-based selection decision. Following the product selection, the software still requires on-going monitoring to ensure it remains healthy, actively developed and adequately supported throughout its lifecycle. This is crucial for identifying and mitigating any risks that emerge as a result of a decline in the project's quality indicators in a timely manner.

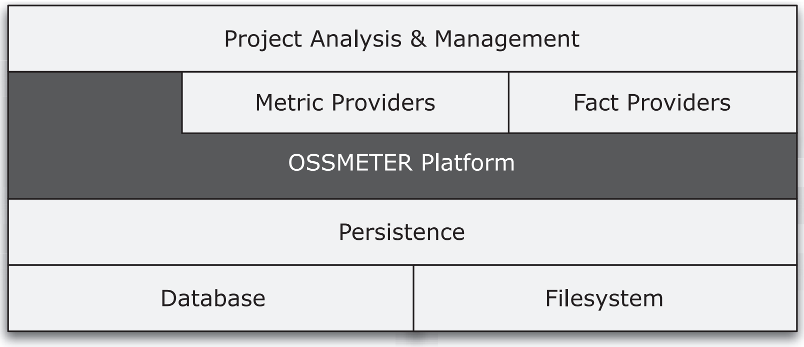


Figure 1: The tiered architecture of the OSSMETER system.

OSSMETER is a Specific Targeted Research Project (STREP) under the Seventh Framework Programme for research and technological development (FP7). The project began in October 2012 and will end in March 2015. A number of different European organizations are involved in the project: The Open Group, University of York and University of Manchester (United

Kingdom), CWI (Netherlands), University of L'Aquila (Italy), Technalia (Spain), Softeam (France), and Uninova and Unparallel Innovation (Portugal). OSSMETER aims to: extend the state-of-the-art in the field of automated analysis and the measurement of OSS; and develop a platform that will support decision makers in the process of discovering, comparing, assessing and monitoring the health, quality, impact and activity of OSS. To achieve these goals OSSMETER will develop trustworthy quality indicators by providing a set of analysis and measurement components. More specifically, the OSSMETER platform will provide the following software measurement components [2]:

- programming language-agnostic and language-specific components to assess a project's source code quality;
- text mining components to analyse the natural language information extracted from communication channels and bug-tracking systems and thus, provide information about communication quality within the project and community activity (around the project); and
- software forge mining components to extract project-related metadata.

After this data is extracted by the platform, it will be made available via a public web application, along with its decision support tools and visualizations.

With almost six months to go, the OSSMETER team is planning to release a beta-version of the platform to the project's industrial partners to support the analysis of OSS projects hosted on SourceForge, GitHub and Eclipse forges. The OSSMETER platform is scheduled to be publicly available at the beginning of 2015. The OSSMETER team will run a public version of the platform which will analyse a number of OSS projects and the results will be published on the OSSMETER website. Furthermore, the platform will also be available for download for personal analyses.

Link:
OSSMETER project:
http://www.ossmeter.org/

Reference:

[1] N. Matragkas et al.: "OSSMETER D5.1 - Platform Architecture Specification", Technical Report, pp. 1-32, Department of Computer Science, University of York, York, UK, 2013 (http://www.ossmeter.eu/publications)

Please contact:

Nicholas Matragkas
University of York
Tel: +44 (0)1904 325164
E-mail: nicholas.matragkas@york.ac.uk

Monitoring Services Quality in the Cloud

by Miguel Zuñiga-Prieto, Priscila Cedillo, Javier Gonzalez-Huerta, Emilio Insfran and Silvia Abrahão

Due to the dynamic nature of cloud computing environments, continuous monitoring of the quality of cloud services is needed in order to satisfy business goals and enforce service-level agreements (SLAs). Current approaches for SLA specifications in IT services are not sufficient since SLAs are usually based on templates that are expressed in a natural language, making automated compliance verification and assurance tasks difficult. In such a context, the use of models at runtime becomes particularly relevant: such models can help retrieve data from the running system to verify SLA compliance and if the desired quality levels are not achieved, drive the dynamic reconfiguration of the cloud services architecture.

Cloud computing represents much more than an infrastructure with which organizations can quickly and efficiently provision and manage their computing capabilities. It also represents a fundamental shift in how cloud applications need to be built, run and monitored. While some vendors are offering different technologies, a mature set of development tools that can facilitate cross-cloud development, deployment and evaluation is yet to be developed. This definitely represents a growth area in the future. The different nature of cloud application development will drive changes in software development process frameworks, which will become more self-maintained and practice-oriented.

Cloud services need to comply with a set of contract clauses and quality requirements, specified by an SLA. To support the fulfillment of this agreement a monitoring process can be defined which allows service providers to determine the actual quality of cloud services. Traditional monitoring technologies are restricted to static and homogenous environments and, as such, cannot be appropriately applied to cloud environments [3]. Further, during the development of these technologies, many assumptions are realized at design time. However, due to the dynamic nature of cloud computing, meeting those assumptions in this context is not possible. It is necessary, therefore, to monitor the continuous satisfaction of the functional and quality requirements at runtime.

During this monitoring process, the violation of an SLA clause may trigger the dynamic reconfiguration of the existing cloud services architecture. Dynamic reconfiguration creates and destroys architectural elements instances at runtime: this is particularly important in the context of cloud computing as their services must continue working while the reconfiguration takes place. However, little attention has been paid to supporting this reconfiguration at runtime and only recently has the field of software engineering research started focusing on these issues [1].

Through the Value@Cloud project, funded by the Ministry of Economy and Competitiveness in Spain, we are developing a framework to support model-driven incremental cloud service development. Specifically, the framework supports cloud development teams to: i) capture business goals and Quality-of-Service (QoS) attributes (which will form part of the SLA); ii) create and incrementally deploy architectural-

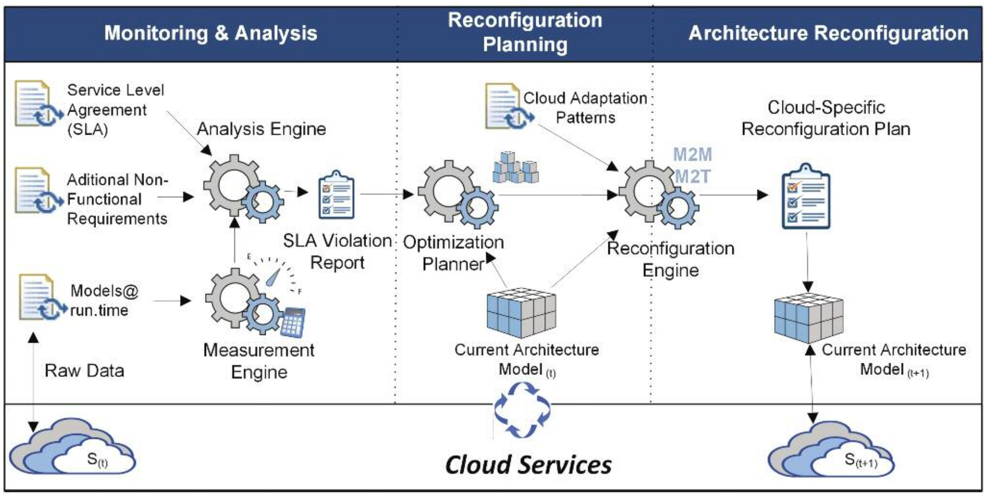

Monitoring & Analysis
Reconfiguration Planning
Architecture Reconfiguration
Service Level Agreement (SLA)
Analysis Engine
Aditional Non-Functional Requirements
SLA Violation Report
Models@run.time
Raw Data
Measurement Engine
Cloud Adaptation Patterns
Optimization Planner
Current Architecture Model (t)
M2M
M2T
Reconfiguration Engine
Cloud-Specific Reconfiguration Plan
Current Architecture Model (t+1)
S(t)
Cloud Services
S(t+1)

Figure 1: Cloud services quality monitoring and reconfiguration infrastructure.

centric cloud services that are capable of dynamically evolving; and iii) monitor the quality of cloud services delivered to the customers.

The monitoring strategy developed through this project is based on two key elements. The first is models at runtime [2] which verify the degree of compliance against the quality requirements specified in the SLA. The second is techniques for dynamically reconfiguring the cloud services architecture if the desired quality levels are not satisfied. The main activities and artifacts involved in this monitoring strategy are shown in Figure 1.

Models at runtime offer flexibility to the monitoring infrastructure through their reflection mechanisms: the modification of quality requirements may dynamically change the monitoring computation, thus avoiding the need to adjust the monitoring infrastructure. In our approach, models at runtime are part of a monitoring & analysis middleware that interacts with cloud services. This middleware retrieves data in the model at runtime, analyzes the information and provides a report outlining the SLA violations. This report is used in the reconfiguration planning to dynamically reconfigure the cloud services architecture in order to satisfy the SLA. The architecture reconfiguration is carried out by generating cloud specific reconfiguration plans, which include adaptation patterns to be applied to cloud service instances at runtime.

We believe that our approach will facilitate the monitoring of the higher-level quality attributes specified in SLAs. It can also provide the architect with flexibility if new quality requirements need to be added or modified since the changes will be performed at runtime and the monitoring infrastructure will remain unchanged. Finally, not only does this approach report the SLA violations identified but also provides a reconfiguration plan for dynamically changing the cloud service architecture in order to satisfy the SLA quality requirements.

Link:
ISSI Research Group at Universitat Politècnica de València:
http://issi.dsic.upv.es/projects

References:

[1] L. Baresi, C. Ghezzi: "The Disappearing Boundary Between Development-time and Run-time", FSE/SDP Workshop on Future of software Engineering Research, pp. 17-21, 2010
[2] N. Bencomo et al.: "Requirements reflection: requirements as runtime entities", 32nd International Conference on Software Engineering, pp. 199-202, 2010
[3] V.C. Emeakaroha et al.: "Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments", HPCS 2010, pp. 48-54.

Please contact:

Silvia Abrahão
Universitat Politècnica de València, Spain
Email: sabrahao@dsic.upv.es

Dictō: Keeping Software Architecture Under Control

by Andrea Caracciolo, Mircea Filip Lungu and Oscar Nierstrasz

Dictō is a declarative language for specifying architectural rules that uses a single uniform notation. Once defined, the rules can automatically be validated using adapted off-the-shelf tools.

Quality requirements (e.g., performance or modifiability) and other derived constraints (e.g., naming conventions or module dependencies) are often described in software architecture documents in the form of rules. For example:

- "Repository interfaces can only declare methods named 'find*()'"; or
- "If an exception is wrapped into another one, the wrapped exception must be referenced as the cause"; and
- "Entity bean attributes of type 'Code' must be annotated with @Type(type = "com.[..].hibernate.CodeMapping")".

Ideally, rules such as these should be checked periodically and automatically. However, after interviewing and surveying dozens of practitioners [1], we discovered that in approximately 60% of cases, these architectural rules are either checked using non-automated techniques (e.g., code review or manual testing) or not checked at all. This situation arises because the automated tools currently available are highly specialized and not always convenient to use. Typically, these tools only handle one kind of rule based on various (often undocumented) theoretical and operational assumptions that hinder their adoption. For a practitioner to be able to validate all their architectural rules, they would need to learn about multiple automated tools, conduct experimental evaluations and set up a proper testing environment. This process requires a significant time and resource investment with no evident payoff.

Our approach in a nutshell

Our goal is to enable practitioners to automatically check that their software systems are evolving without straying from previously established architectural rules. To support this idea, we propose that Dictō [2], a unified DSL (domain specific language), could be used to specify the architectural rules, thus allowing them to be automatically tested using adapted off-the-shelf tools.

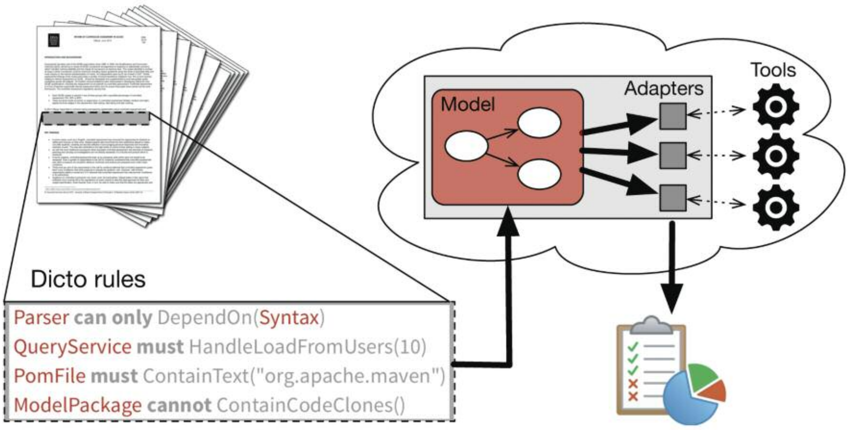
This proposed approach supports a fully automated verification process, allowing even non-technical stakeholders to be involved in the specification of rules. Once the rules are expressed, their verification can be integrated into the continuous integration system. This ensures the correct implementation of the planned architecture over time and helps prevent architectural decay.

How it works

Using Dictō, an architectural rule such as “The web service must answer user requests within 10ms”can be expressed as “WebService must HaveResponseTime LessThan (“10 ms”)”.

Rules are composed of subject entities and logical predicates. In this example, the subject (“WebService”) is an entity defined by the user, which maps to a concrete element of the system. The predicate (“HaveResponseTimeLessThan”) is formulated to prescribe the expected properties on the specified subjects. To increase expressivity without sacrificing readability, we support four types of rules: must, cannot, only-can and can-only.

Dictō rules are parsed and fed to the most appropriate tool through purpose-built adapters (Figure 1). These are plug-ins designed to accept rules that match a specific syntactic structure. The accepted rules are then analyzed and used to generate a valid input specification for the adapted tool. The results obtained from each of the supported tools can eventually be aggregated and used to build an overall report for the user. The adapters are written by tool experts and, by contributing the necessary code to the Dictō project, can be shared with a larger user-base.



```
Parser can only DependOn(Syntax)
QueryService must HandleLoadFromUsers(10)
PomFile must ContainText("org.apache.maven")
ModelPackage cannot ContainCodeClones()
```

Figure 1: Our approach, from the specification of the rules to their testing.

Implementation

The current version of Dictō is capable of testing rules defined over various aspects of a system from observable behaviours (e.g., latency, load or uptime) to implementation (e.g., dependencies or code clones). It features adapters for six different tools (JMeter, JavaPathFinder, PMD, grep, Moose and ping). By collaborating with interested stakeholders, we plan to extend this catalogue of testable rules. This future step will offer invaluable insights into the needs of actual users.

This work has been developed by Andrea Caracciolo (Software Composition Group, University of Bern). It is part of the “Agile Software Assessment” project funded by the Swiss National Science Foundation. The tool is freely available for download on our website (http://scg.unibe.ch/dicto-dsl/) and the source code is open under the MIT license. We are interested in both academic and industry collaborations to further develop and assess our approach.

Link:
http://scg.unibe.ch/dicto-dsl/

References:

[1] A. Caracciolo, M. F. Lungu, and O. Nierstrasz: “How Do Software Architects Specify and Validate Quality Requirements?”, in Software Architecture, LNCS 8627, p. 374-389, Springer, 2014

[2] A. Caracciolo, M. F. Lungu, and O. Nierstrasz: “Dicto: A Unified DSL for Testing Architectural Rules”, in ECSAW ‘14, ACM, 2014.

Please contact:

Andrea Caracciolo
University of Bern, Switzerland
E-mail: caracciolo@iam.unibe.ch

Dedicated Software Analysis Tools

by Nicolas Anquetil, Stéphane Ducasse and Usman Bhatti

The data and software analysis platform Moose allows for the quick development of dedicated tools that can be customized at different levels. These tools are crucial for large software systems that are subject to continuous evolution.

The lifetime of large systems (such as those that support the activities of banks, hospitals, insurance companies and the army) can be measured in decades. Such software systems have become a crucial component for running the day-to-day affairs of our society. Since these systems model important aspects of human activity, they must undergo continuous evolution that follows the evolution of our society. For example, new laws, economical constraints or requirements force large software systems to evolve. Previous studies have shown that undertaking this evolution can represent up to 90% of total software effort [1]. Controlling such systems and ensuring they can evolve is a key challenge: it calls for a detailed understanding of the system, as well as its strengths and weaknesses. Deloittes recently identified this issue as an emerging challenge [2].

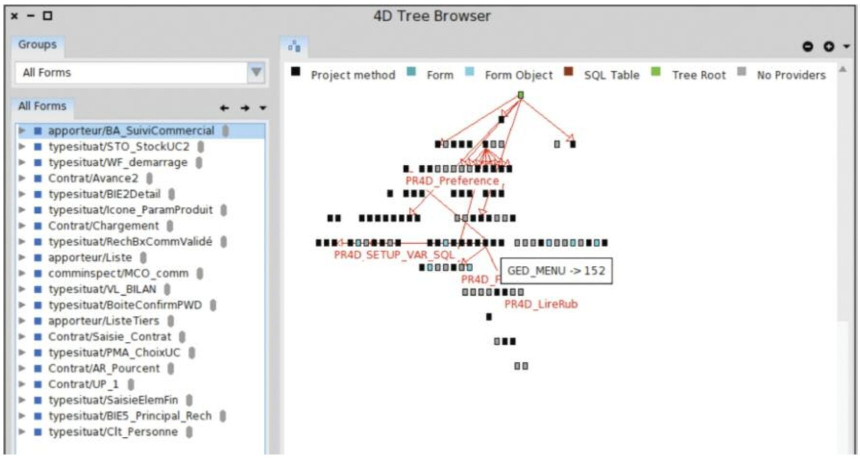
From an analysis of the current situation, four key facts emerge.

1. Despite the importance of software evolution for our economy, it is not considered to be a relevant problem (and in fact, is considered a topic of the past): for example, currently there are no EU research axes that focus on this crucial point while buzzwords such as "big data" and "the Cloud" attract all the attention.
2. People seem to believe that the issues associated with software analysis and evolution have been solved, but the reality is little has been accomplished.
3. New development techniques such as Agile Development, Test Driven Development, Service-Oriented Architecture and Software Product Lines cannot solve the problems that have accumulated over years of maintenance on legacy systems and it is impossible to dream of redeveloping even a small fraction of the enormous quantity of software that currently exists today.
4. Software evolution is universal: it happens to any successful software, even in projects written with the latest and coolest technologies. The productivity increases that have been achieved with more recent technologies will further complicate the issue as engineers produce more complex code that will also have to be maintained. There are tools that propose some basic analyses in terms of the "technical debt" (i.e., that put a monetary value on bad code quality), however, knowing that you have a debt does not help you take action to improve code quality.

Typical software quality solutions that assess the value of some generic metrics at a point in time are not adapted to the needs of the developers. Over the years we have developed Moose, a data and software analysis platform. We have previously presented Moose [3], but in this article, we want to discuss some of the aspects we learnt while selling tools built on top of Moose. In conjunction with the clients of our associated company Synectique, we identified that an adequate analysis infrastructure requires the following elements.

The first is *dedicated processes and tools* which are needed to approach the specific problems a company or system might face. Frequently software systems use proprietary organization schemes to complete tasks, for example, to implement a specific bus communication between components. In such cases, generic solutions are mostly useless as they only give information in terms of the "normal", low-level concepts available in the programming language used. Large software systems need to be analyzed at a higher abstraction level (e.g., component, feature or sub-system). This supports reverse engineering efforts. In Moose, we offer a meta model-based solution where the imported data is stored independently of the programming language. This approach can be extended to support proprietary concepts or idioms, and new data can be supported by merely adapting the model and defining the proper importer. Once the information is imported, analysts can take advantage of the different tools for crafting software analyses that are tailored to meet their needs.

The second element is *tagging*. End users and/or reengineers often require a way to annotate and query entities with expert knowledge or the results of an analysis. To respond to this need, end users and reengineers are provided with a tagging mechanism which allows them to identify interesting entities or their groups. An interesting case which highlights the use of this mechanism is the extraction of a functional architecture from structural model information. Once experts or analyses have tagged entities, new tools and analyses (such as a rule-based validation) use it (by querying) to advanced knowledge and create more results.

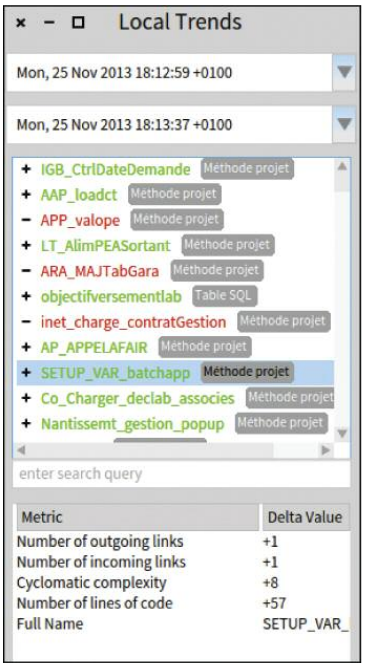


apporteur/BA_SuiviCommercial
typesituat/STO_StockUC2
typesituat/WF_demarrage
Contrat/Avance2
typesituat/BIE2Detail
typesituat/Icone_ParamProduit
Contrat/Chargement
typesituat/RechBxCommValidé
apporteur/Liste
comminspect/MCO_comm
typesituat/VL_BILAN
typesituat/BoiteConfirmPWD
apporteur/ListeTiers
Contrat/Saisie_Contrat
typesituat/PMA_ChoixUC
Contrat/AR_Pourcent
Contrat/UP_1
typesituat/SaisieElemFin
typesituat/BIE5_Principal_Rech
typesituat/Clt_Personne

Figure 1: A dependency analyzer for legacy code.

The third element is the *dependency nightmare* analysis and remediation tool. Large and/or old software systems have often suffered from architectural drift for so long that there is little or no architecture left. All parts of a software system are intrinsically linked and frequently, loading three modules can mean loading the complete system. The challenge is how to deal with this fine grained information at a large grain level. We propose an advanced cyclic dependency analysis and removal tools as well as a drill-down on architectural views. Figure 1 shows the tool revealing recursive dependencies for impact analysis.

The fourth element is a *trend analysis* solution. Instead of a punctual picture of a system's software state, it is desirable to understand the evolution of the quality of entities. As the source code (and thus the software entities) typically evolve in integrated development environments, independently from the dedicated, off-the-shelf, software quality tools, computing quality analysis trends requires changes (e.g., add, remove, move or rename) of individual software entities identified. We propose a tool that computes such changes and the metric evolutions. Figure 2 shows the changes computed on two versions (green: entity added, red: entity removed) and the evolution of quality metrics for a change. Queries may be expressed in terms of the changes (i.e., “all added methods”) or in terms of the metric variations (i.e., “increase of CyclomaticComplexity > 5”).


Local Trends

Mon, 25 Nov 2013 18:12:59 +0100

Mon, 25 Nov 2013 18:13:37 +0100

- + IGB_CtrlDateDemande Méthode projet
- + AAP_loadct Méthode projet
- − APP_valope Méthode projet
- + LT_AlimPEASortant Méthode projet
- − ARA_MAJTabGara Méthode projet
- + objectifversementlab Table SQL
- − inet_charge_contratGestion Méthode projet
- + AP_APPELAFAIR Méthode projet
- + SETUP_VAR_batchapp Méthode projet
- + Co_Charger_declab_associes Méthode projet
- + Nantissemt_gestion_popup Méthode projet

enter search query

Metric	Delta Value
Number of outgoing links	+1
Number of incoming links	+1
Cyclomatic complexity	+8
Number of lines of code	+57
Full Name	SETUP_VAR_

Figure 2: Trends analysis.

Conclusion

Software evolution and maintenance is, and will continue to be, a challenge for the future. This is not because a lack of research advances but rather because more and more software is being created and that software is destined to last longer. In addition, any successful software systems must evolve to adapt to global changes. Our experience shows that while problems may look similar on the surface, key problems often require dedicated attention (e.g., processing, analyses and tools). There is a need for dedicated tools that can be customized at different levels, such as models, offered analyses and the level of granularity.

Links:
http:// www.moosetechnology.org
http://www.synectique.eu

References:

[1] C. Jones: “The Economics of Software Maintenance in the Twenty First Century”, 2006, http://www.compaid.com/caiinternet/ezine/capersjones-maintenance.pdf

[2] B. Briggs et al.: “Tech Trends 2014, Inspiring Disruption”, White Paper, Deloitte University Press, 2014, http://www.deloitte.com/assets/Dcom-Luxembourg/Local%20Assets/Documents/Whitepapers/2014/dtt_en_wp_techtrends_10022014.pdf

[3] O. Nierstrasz, S. Ducasse, T. Gîrba: “The story of Moose: an agile reengineering environment”, ESEC/SIGSOFT FSE 2005: 1-10.

Please contact:

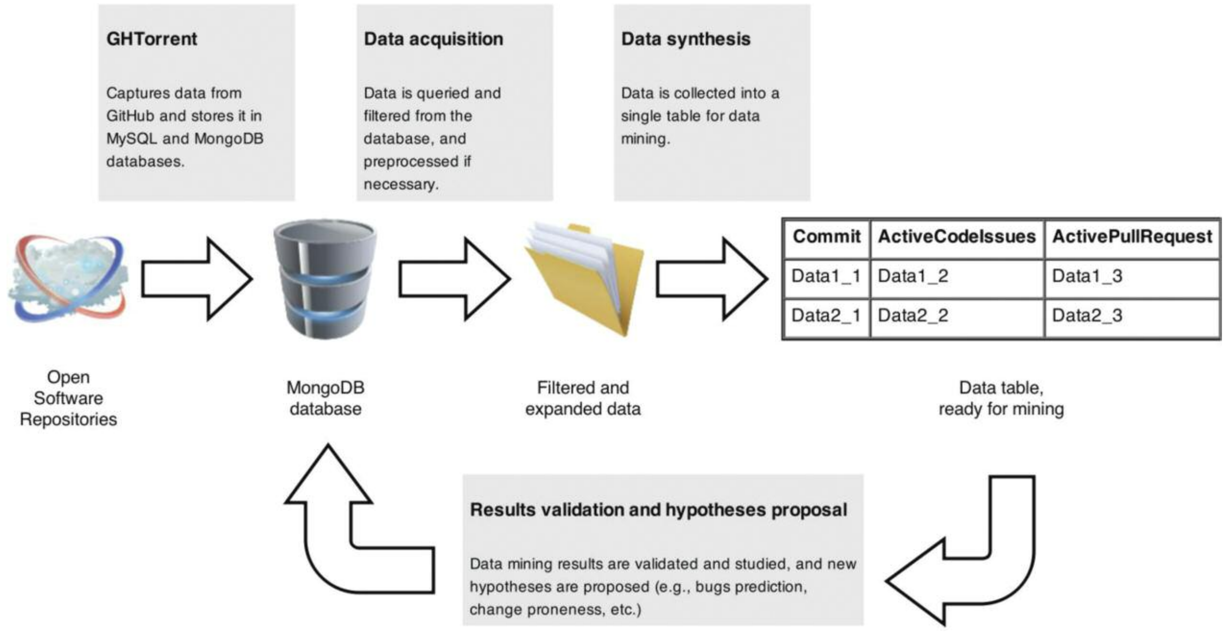
Stéphane Ducasse
Inria, France
E-mail: stephane.ducasse@inria.fr

Mining Open Software Repositories

by Jesús Alonso Abad, Carlos López Nozal and Jesús M. Maudes Raedo

With the boom in data mining which has occurred in recent years and higher processing powers, software repository mining now represents a promising tool for developing better software. Open software repositories, with their availability and wide spectrum of data attributes are an exciting testing ground for software repository mining and quality assessment research. In this project, the aim was to achieve improvements in software development processes in relation to change control, release planning, test recording, code review and project planning processes.

In recent years, scientists and engineers have started turning their heads towards the field of software repository mining. The ability to not only examine static snapshots of software but also the way they have evolved over time is opening up new and exciting lines of research towards the goal of enhancing the quality assessment process. Descriptive statistics (e.g., mean, median, mode, quartiles of the data-set, variance and standard deviation) are not enough to generalize specific behaviours such as how prone a file is to change [1]. Data mining analysis (e.g., clustering, regression, etc.) which are based on the newly accessible information from software repositories (e.g., contributors, commits, code frequency, active issues and active pull requests) must be developed with the aim of proactively improving software quality, not only reactively responding to issues. Open source software repositories like Sourceforge and GitHub provide a rich and varied source of data to mine. Their open nature welcomes contributors with very different skill sets and experience levels and the absence or low levels of standardized workflow enforcement make them reflect ‘close-to-extreme’ cases (as opposed to the more structured


GHTorrent
Captures data from GitHub and stores it in MySQL and MongoDB databases.
Data acquisition
Data is queried and filtered from the database, and preprocessed if necessary.
Data synthesis
Data is collected into a single table for data mining.
Open Software Repositories
MongoDB database
Filtered and expanded data
Data table, ready for mining
Results validation and hypotheses proposal
Data mining results are validated and studied, and new hypotheses are proposed (e.g., bugs prediction, change proneness, etc.)

Commit	ActiveCodeIssues	ActivePullRequest
Data1_1	Data1_2	Data1_3
Data2_1	Data2_2	Data2_3

Figure 1: The process of mining data from an open software repository.

workflow patterns experienced when using, for instance, a branch-per-task branching policy). In addition, they provide easily accessible data sources for scientists to experiment with. The collection of these massive amounts of data have been supported by Qualitas Corpus [2] and GHTorrent [3] who have both made multiple efforts to gather and offer datasets to the scientific community.

The project workflow, undertaken by our research team at the University of Burgos, Spain, included the following steps (Figure 1):

1. Obtain data collected by GHTorrent from the GitHub repository and put it into MongoDB databases.
2. Filter the data according to needs and expand the data where possible (e.g., downloading source code files or calculating measurements such as the number of commits, number of issues opened, etc.). Some pre-processing of the data using JavaScript was completed during the database querying step and a number of Node.js scripts were used for several operations afterwards (e.g., file downloading or calculating static code metrics such as the number of lines of code, McCabe's complexity, etc.)
3. Define an experiment with the aim of improving the software development process and pack the expanded data into a data table that will be supplied to a data mining tool to be used for a range of different techniques including regression or clustering.
4. Evaluate the data mining results and prepare experiments to validate new hypotheses based on those results.

Despite the benefits of using such repositories, it is important to remember that, sometimes, a lack of standarization in the integration process can create unformatted or missing commit messages or frequent unstable commits. This, and other constraints (not discussed here) can make data mining these repositories more difficult and/or lead to sub-optimal results.

Until now, software quality assessment has focused on single snapshots taken throughout the life of the software. Thus, the assessments have not been able to take the time variable into account. The use of software repositories allows researchers to address this shortcoming. Consequently, future software repository mining will play a key role in enhancing the software development process, allowing developers to detect weak points, predict future issues and provide optimized processes and development cycles. Open software repositories offer a number of future research opportunities.

Links:
http://sourceforge.net/
https://github.com/
http://qualitascorpus.com/
http://ghtorrent.org/

References:

[1] I. S. Wiese et al.: "Comparing communication and development networks for predicting file change proneness: An exploratory study considering process and social metrics," Electron. Commun. EASST - proc. of SQM 2014, vol. 65, 2014.
[2] E. Tempero et al.: "The Qualitas Corpus: A Curated Collection of Java Code for Empirical Studies," 2010 Asia Pacific Softw. Eng. Conf., 2010.
[3] G. Gousios, D. Spinellis: "GHTorrent: Github's data from a firehose," 9th IEEE MSR, 2012.

Please contact:

Jesús Alonso Abad
University of Burgos, Spain
Tel: +34 600813116
E-mail: jesus.alonso.abad@gmail.com

Carlos López Nozal
University of Burgos, Spain
Tel: +34 947258989
E-mail: clopezno@ubu.es

Jesús M Maudes Raedo.
University of Burgos, Spain
Tel: +34 947259358
E-mail: jmaudes@ubu.es

A Refactoring Suggestion Tool for Removing Clones in Java Code

by Francesca Arcelli Fontana, Marco Zanoni and Francesco Zanoni

Code duplication is considered a widespread code smell, a symptom of bad code development practices or potential design issues. Code smells are also considered to be indicators of poor software maintainability. The refactoring cost associated with removing code clones can be very high, partly because of the number of different decisions that must be made regarding the kind of refactoring steps to apply. Here, we describe a tool that has been developed to suggest the best refactoring steps that could be taken to remove clones in Java code. Our approach is based on the classification of clones, in terms of their location in a class hierarchy, so that decisions can be made from a restricted set of refactorings that have been evaluated using multiple criteria.

Code clones [1] have been widely studied and there is a large amount of literature on this issue. This work has led to a number of different types of clones being identified. Clones involve all non-trivial software systems; the percentage of involved duplicated lines is usually estimated to be between 5% and 20% but can sometimes even reach 50% [2]. Many of the studies have investigated the factors that cause clone insertion and their results have enabled several criteria and detection techniques to be developed.

When addressing the issue of duplicated code management, we have to consider the following aspects:

- what instances are worth refactoring and which are not; and
- once an instance has been evaluated as worth of refactoring, which technique should be applied to remove a duplicated instance.

Refactoring duplicated code is a task in which code fragments are merged or moved to other locations, for example other functions, methods or classes. Moving code means that the computational logic belonging to a specific entity of the system is moved: it should be approached with caution as relocation can break the original design coherence, reducing cohesion and/or moving responsibilities to unsuitable entities. There are a number of refactoring techniques available, each having its own pros and cons in both design and lower-level aspects.

In this study, we proposed an approach that aims at automatically evaluating and selecting suitable refactoring techniques based on the classification of the clones, thus reducing the human involvement in the process. We focused our attention on the following aspects:

- an analysis of the location of each clone pair resulting in a specific set of applicable refactoring techniques,
- the ranking of the applicable refactoring techniques based on a set of weighting criteria, and
- the aggregation of the critical clone information and best refactoring techniques, according to those numerical criteria.

In line with this vision, we developed a tool which suggests the ‘best’ refactoring techniques for code clones in Java and named it the Duplicated Code Refactoring Advisor (DCRA; Figure 1). The tool consists of four components, each designed with a specific goal. Every component enriches the information obtained on the duplicate code and the whole elaboration process identifies a suitable list of techniques that could be applied to the most problematic duplications. The four components are:

- the *Clone Detector*, which is an external tool for detecting clone pairs (we are currently using a well known tool called NiCad [3]);
- the *Clone Detailer*, which analyzes the Clone Detector output and characterises every clone, detailing information such as clone location, size and type;
- the *Refactoring Advisor*, which visits a decision tree to choose the possible refactoring techniques related to each clone pair; the use of this component allows for refactoring technique suggestions to be made, based on the clone location and the variables contained in the clone; suggestions are ranked on the basis of the clone’s different features, e.g., a Lines of Code (LOC) variation and an evaluation of the quality resulting from its application, in terms of object-oriented programming constructs exploitation; and
- the *Refactoring Advice Aggregator*, which aggregates the available information on clones and refactoring techniques, groups them by class or package and then sorts them by refac-

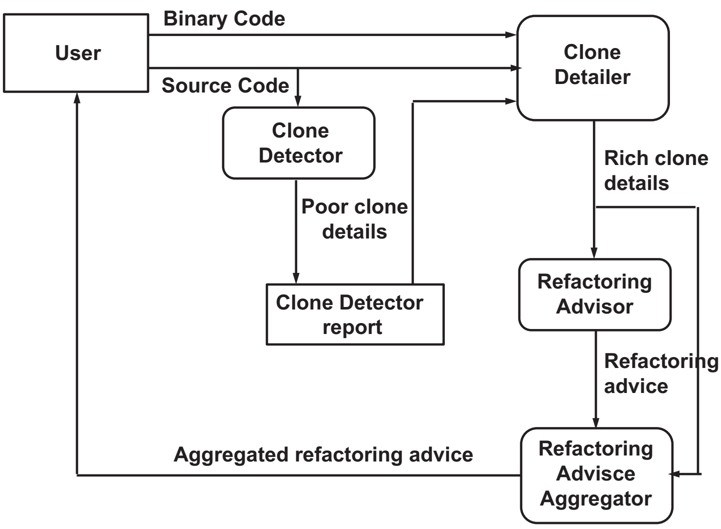


Figure 1: Duplicate code data flow through DCRA components.

toring significance or clone pair impact, thus providing a summary report which captures the most interesting information about clones, e.g., what are the largest clones and which clones should be easiest (or most convenient) to remove.

In developing this approach, our dual aim was to filter out which clone pairs are worthy of refactoring and suggest the best refactoring techniques for those worthy clone pairs. We have successfully provided an automated technique for selecting the best refactoring techniques in a given situation that is based on a classification of code clones. We experimented the Clone Detailer module on 50 systems of the Qualitas Corpus from Tempero et al. We validated all the modules of our DCRA tool on four systems of the Qualitas Corpus. The tool suggested a successful refactoring in most cases.

Through its use, the aim is that DCRA will offer a concrete reduction in the human involvement currently required in duplicated code refactoring procedures and, thus, reducing the overall effort required from software developers.

References:

[1] M. Fowler: "Refactoring. Improving the Design of Existing Code", Addison- Wesley, 1999

[2] M. F. Zibran, C. K. Roy: "The road to software clone management: A survey", The Univ. of Saskatchewan, Dept. Computer Science, Tech. Rep. 2012-03, Feb. 2012, http://www.cs.usask.ca/documents/techreports/2012/TR- 2012- 03.pdf

[3] C. Roy, J. Cordy: "NICAD: Accurate detection of near-miss intentional clones using flexible pretty-printing and code normalization", in proc. of ICPC 2008, Amsterdam, pp. 172–181.

Please contact:

Francesca Arcelli Fontana,
Marco Zanoni
University of Milano Bicocca, Italy,
E-mail: arcelli@disco.unimib.it,
zanoni@disco.unimib.it

Debugging with the Crowd: A Debug Recommendation System Based on StackOverflow

by Martin Monperrus, Anthony Maia, Romain Rouvoy and Lionel Seinturier

Leveraging the wisdom of the crowd to improve software quality through the use of a recommendation system based on StackOverflow, a famous Q&A website for programmers.

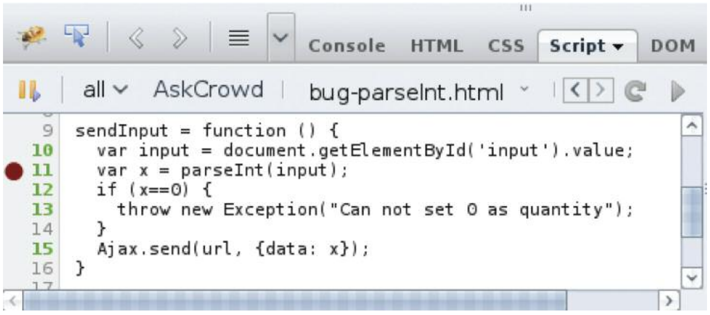
Debugging continues to be a costly activity in the process of developing software. However, there are techniques to decrease overall the costs of bugs (i.e., bugs get cheaper to fix) and increase reliability (i.e., on the whole, more bugs are fixed).

Some bugs are deeply rooted in the domain logic but others are independent of the specificity of the application being debugged. This latter category are called "crowd bugs", unexpected and incorrect behaviours that result from a common and intuitive usage of an application programming interface (API). In this project, our research group here at Inria Lille (France), set out to minimize the difficulties associated with fixing crowd bugs. We propose a novel debugging approach for crowd bugs [1]. This debugging technique is based on matching the piece of code being debugged against related pieces of code reported by the crowd on a question and answer (Q&A) website.

To better define what a "crowd bug" is, let us first begin with an example. In JavaScript, there is a function called parseInt, which parses a string given as input and returns the corresponding integer value. Despite this apparently simple description and self-described signature, this function poses problems to many developers, as witnessed by the dozens of Q&As on this topic (http://goo.gl/m9bSJS). Many of these specially relate to the same question, "Why does parseInt("08") produce a '0' and not an '8'?" The answer is that if the argument of parseInt begins with 0, it is parsed as an octal. So why is the question asked again and again? We hypothesize that the semantics of parseInt are counter-intuitive for many people and consequently, the same issue occurs over and over again in development situations, independently of the domain. The famous Q&A website for programmers, StackOverflow (http://stackoverflow.com), contains thousands of crowd bug Q&As. In this project, our idea was to harness the information contained in the coding scenarios posed and answered on this website at debugging time.

The new debugger we are proposing works in the following manner. When faced with an understandable bug, the developer sets a breakpoint in the code presumed to be causing the bug and then approaches the crowd by clicking the 'ask the crowd' button The debugger then extracts a 'snippet', which is defined as n lines of code around the breakpoint, and cleans them according to various criteria. This snippet acts as a query which is then submitted to a server which in turn, retrieves a list of Q&As that match that query. The idea is that within those answers lies a range of potential solutions to the bug and allows for those solutions to be reused. The user interface of our prototype is based on a crowd-based extension of Firebug, a JavaScript debugger for Firefox (Figure 1).

To determine the viability of this approach, our initial task was to confirm whether Q&A websites such as StackOverflow were able to handle snippet inputs well. Our approach only uses code to query the Q&As, as opposed to text elaborated by devel-



```
 9  sendInput = function () {
10    var input = document.getElementById('input').value;
11    var x = parseInt(input);
12    if (x==0) {
13      throw new Exception("Can not set 0 as quantity");
14    }
15    Ajax.send(url, {data: x});
16  }
```

Figure 1: A screenshot of our prototype crowd-enhanced JavaScript debugger. The button 'AskCrowd' selects the snippet surrounding the breakpoint (Line 11: red circle). A list of selected answers that closely match the problem are then automatically retrieved from StackOverflow. In this case, the first result is the solution: the prefix has a meaning for parseInt which must be taken into account.

opers. To investigate this question, we took a dataset that comprised of 70,060 StackOverflow Q&As that were determined to possibly relate to JavaScript crowd-bugs (dataset available on request). From this dataset, 1000 Q&As and their respective snippets were randomly extracted. We then performed 1,000 queries to the StackOverflow search engine, using the snippets only as an input. This analysis yielded the following results: 377 snippets were considered to be non-valid queries, 374 snippets yielded no results (i.e., the expected Q&A was not found) and finally, 146 snippets yield a perfect match (i.e., the expected Q&A was ranked #1).

These results indicate that StackOverflow does not handle code snippets inputted as query particularly well.

In response to this issue, we introduced pre-processing functions aimed at improving matching quality between the snippets being debugged and the ones on the Q&A repository. We experimented with different pre-processing functions and determined that the best one is based on a careful filtering of the abstract syntax tree of the snippet. Thus, using this pre-processing function, we repeated the same evaluation followed above and on this occasion, 511 snippets yielded a #1 ranked Q&A (full results can be found in our online technical report [1]). Consequently, we integrated this pre-processing function into our prototype crowd-bug debugger to maximize the chances of finding a viable solution.

Beyond this case, we are conducting further research which aims to leverage crowd wisdom to improve the automation of software repair and contribute to the overarching objective of achieving more resilient and self-healing software systems.

Links:
https://team.inria.fr/spirals
http://stackoverflow.com

Reference:

[1] M. Monperrus, A. Maia: "Debugging with the Crowd: a Debug Recommendation System based on Stackoverflow", Technical report #hal-00987395, INRIA, 2014.

Please contact:

Martin Monperrus
University Lille 1, France
Inria/Lille1 Spirals research team
Tel: +33 3 59 35 87 61
E-mail:
Martin.Monperrus@univ-lille1.fr

RiVal: A New Benchmarking Toolkit for Recommender Systems

by Alan Said and Alejandro Bellogín

RiVal is a newly released toolkit, developed during two ERCIM fellowships at Centrum Wiskunde & Informatica (CWI), for transparent and objective benchmarking of recommender systems software such as Apache Mahout, LensKit and MyMediaLite. This will ensure that robust and comparable assessments of their recommendation quality can be made.

Research on recommender systems often focuses on making comparisons of their predictive accuracy, i.e., the better the evaluation scores, the better the recommender. However, it is difficult to compare results between different recommender systems and frameworks, or even assess the quality of one system, due to the myriad of design and implementation options in the evaluation strategies. Additionally, algorithm implementations can diverge from the standard formulation due to manual tuning and modifications that work better in some situations. We have developed an open source evaluation toolkit for recommender systems (RiVal), which provides a set of standardised evaluation methodologies. This was achieved by retaining complete control of the evaluation dimensions being benchmarked (i.e., data splitting, metrics, evaluation strategies, etc.), independent of the specific recommendation strategies.

Recommender systems are a popular means of assisting users of a range of online services in areas, such as music (e.g., Spotify, Last.fm), movies and videos (e.g., Netflix, YouTube) or other items (e.g., Amazon, eBay) [1]. In recent years, research in this field has grown exponentially and today most top-tier research venues feature tracks on recommendation. There has been a parallel development in industry and now, many data science positions place a significant emphasis on candidates possessing expertise in recommendation techniques. This gain in popularity has led to an overwhelming growth in the amount

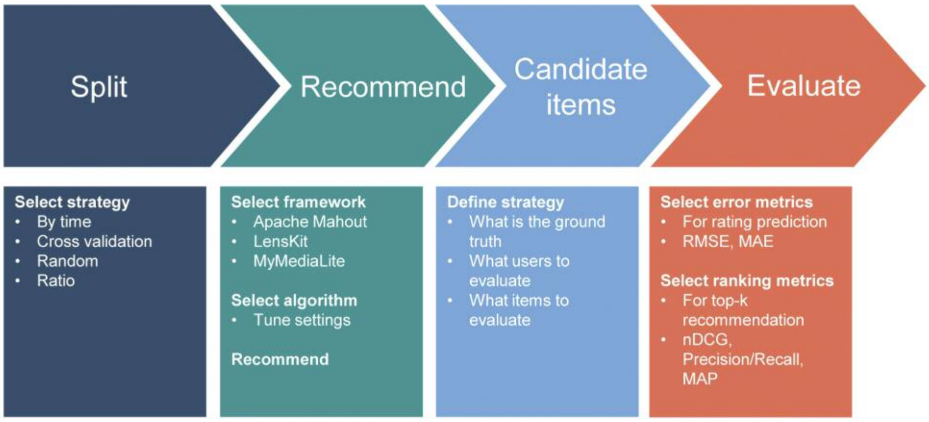


Figure 1: The RiVal evaluation pipeline. The toolkit's modular design means each module can be executed individually (i.e., only the evaluation module or only the data splitting module) or alternatively, the complete pipeline can be executed within RiVal.

of available literature, as well as a large set of algorithms to be implemented. With this in mind, it is becoming increasingly important to be able to benchmark recommendation models against one another to objectively estimate their performance.

Usually, each implementation of an algorithm is associated with a recommendation framework or software library, which in turn, must provide additional layers to access the data, report performance results, etc. An emerging problem associated with having numerous recommendation frameworks is the difficulty in comparing results across software frameworks, i.e., the reported accuracy of an algorithm in one framework will often differ from the same algorithm in a different framework. Minor differences in algorithmic implementation, data management and evaluation are among the number of causes of this problem. To properly analyse this problem, we have developed RiVal, a software toolkit that is capable of efficiently evaluating recommender systems, RiVal can test the various functionalities of recommender systems while simultaneously being agnostic to the actual algorithm in use. It does not incorporate recommendation algorithms but rather, provides bindings or wrappers to the three recommendation frameworks most common at the moment, Apache Mahout, LensKit and MyMediaLite.

RiVal provides a transparent evaluation setting which gives the practitioner complete control of the various evaluation steps. More specifically, it is composed of three main modules: data splitting, candidate item generation and performance measurement. In addition, an item recommendation module is also provided that integrates the three common recommendation frameworks (listed above) into the RiVal pipeline (Figure 1). RiVal is now available on GitHub and further development information can be found on its Wiki and manual pages (see Links section below). The toolkit's features are also outlined in detail in a paper [2] and demo [3]. The toolkit can be used programmatically as Maven dependencies, or by running it as a standalone program for each of the steps.

By using RiVal, we have been able to benchmark the three most common recommendation algorithms implemented in the three aforementioned frameworks using three different datasets. We also generated a large amount of results by using our controlled evaluation protocol, since it consisted of four data splitting techniques, three strategies for candidate item generation, and five main performance metrics. Our results point to a large discrepancy between the same algorithms implemented in different frameworks. Further analyses of these results [2] indicate that many of these inconsistencies were a result of differences in the implementation of the algorithms. However, the implementation of the evaluation metrics and methods also differed across the frameworks, which makes the objective comparison of the recommendation quality across the frameworks impossible when using a framework-internal evaluation.

The RiVal toolkit enables practitioners to perform completely transparent and objective evaluations of recommendation results, which will improve the selection of which recommendation framework (and algorithm) should be used in each situation. Providing an evaluation system, which is highly configurable, unbiased by framework-dependent implementations, and usable across frameworks and datasets, allows both researchers and practitioners to assess the quality of a recommender system in a wider context than current standards in the area allow.

This research was developed while both authors were ERCIM fellows at CWI.

Links:
RiVal: http://rival.recommenders.net
Apache Mahout:
https://mahout.apache.org
LensKit: http://lenskit.org
MyMediaLite:
http://www.mymedialite.net

References:
[1] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor: "Recommender Systems Handbook", Springer, 2011
[2] A. Said, A. Bellogín: "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks", in ACM RecSys, 2014
[3] A. Said, A. Bellogín: "RiVal – A Toolkit to Foster Reproducibility in Recommender System Evaluation", in ACM RecSys, 2014.

Please contact:
Alan Said
TU-Delft, The Netherlands
E-mail: alansaid@acm.org

Alejandro Bellogín
Universidad Autónoma de Madrid, Spain
E-mail: alejandro.bellogin@uam.es

Evaluating the Quality of Software Models using Light-weight Formal Methods

by Jordi Cabot and Robert Clarisó

For non-critical software systems, the use of light-weight formal evaluation methods can guarantee sufficient levels of quality.

Model-Driven Engineering (MDE) is a software engineering paradigm aimed at improving developer productivity and software quality. To this end, the development process in MDE does not focus on the code but rather, on the models of the system being built. Models characterize the relevant features of a system for a specific purpose, for example, documentation, analysis, code generation or simulation. By abstracting irrelevant features, system complexity is reduced and, due to support from (semi)automatic tools for some development tasks, developers can minimise human-introduced errors and enhance productivity.

For this to hold true, model quality should be a primary concern: a defect in a model can propagate into the final implementation of the software system. As for the software, the quality of the models can be regarded from many different perspectives. It is necessary to make sure that the models are realizable (i.e., the structural models should be able to be satisfied, the states in a behavioral model should be reachable, etc.). In addition, models that offer complementary views of the same system should be consistent.

Formal methods provide valuable techniques to ensure the correctness of these software models. Fortunately, abstraction makes models more amenable to analysis than source code. In any case, most formal verification problems are undecidable or have such high computational complexity that it hampers scalability. Thus, software verification is, and will remain, a grand challenge for software engineering research in the foreseeable future [1].

Beyond issues of scale, other model quality challenges include incomplete models and model evolution. These factors make the application of fully-fledged (complete and possibly undecidable) formal methods unsuitable in this context. Instead, light-weight approaches that are able to provide quick feedback and support large and complex models may be preferred, even if their outcomes can sometimes be inconclusive. We believe this pragmatic approach offers the best trade-off for non-critical software systems. In particular, we have been applying a family of light-weight formal methods, based on bounded verification by means of constraint programming, to evaluate the correctness of Unified Modeling Language(UML)/Object Constraint Language (OCL) software models.

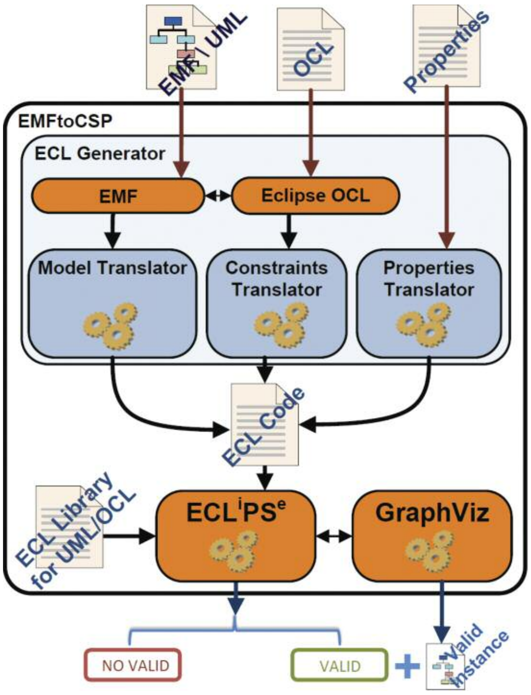
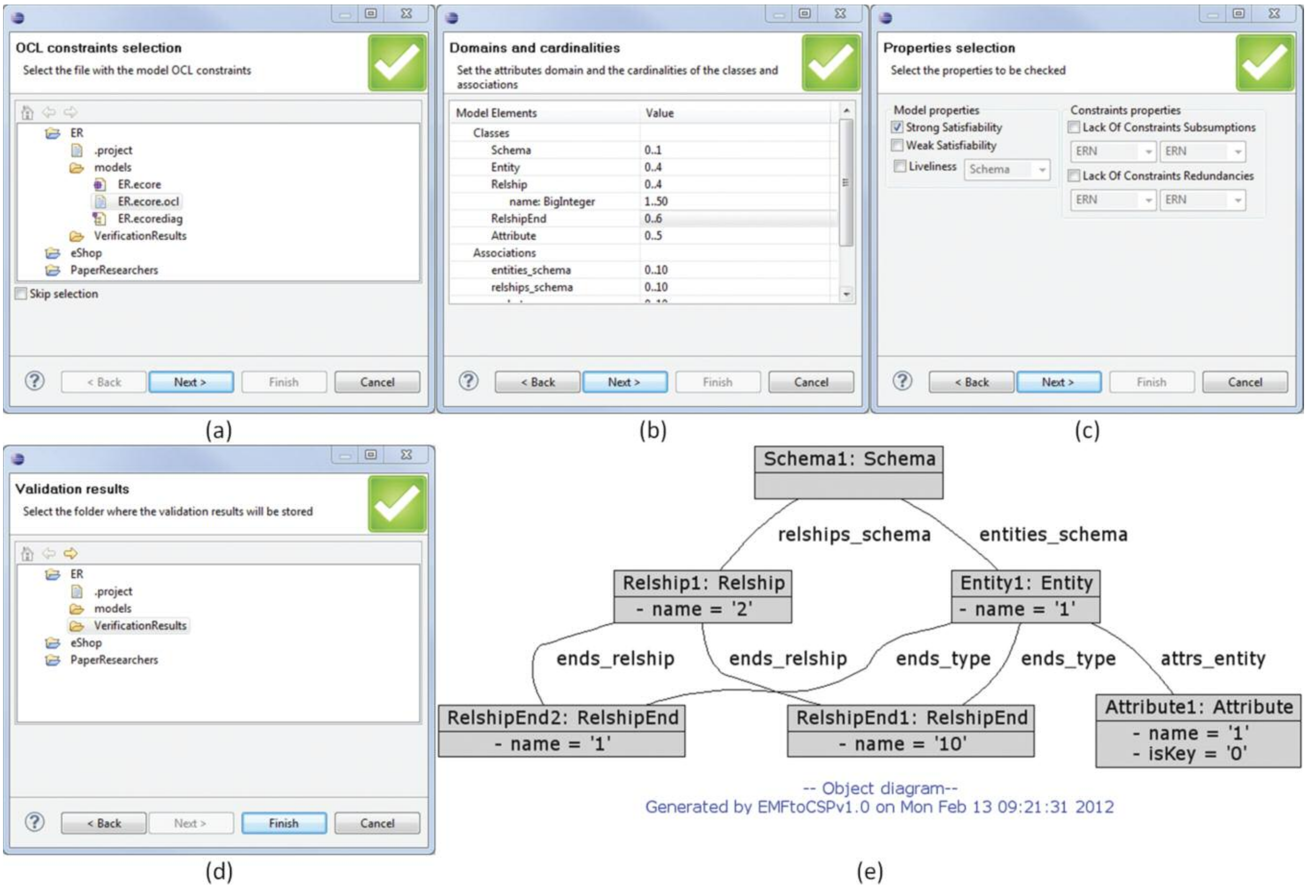


Figure 1: The architecture of the EMFtoCSP tool.

The UML is a de facto standard for describing software models, providing a collection of visual and textual notations to describe different facets of a software system. The most popular UML notation is the class diagram, which depicts classes within an object-oriented hierarchy. However, UML class diagrams are unable to capture complex integrity constraints beyond basic multiplicity constraints. For more advanced constraints, class diagrams can be augmented using a companion textual notation, the OCL.

Using UML/OCL allows complex software systems to be designed without committing to a specific technology or platform. These designs can be verified to detect defects before investing effort in implementation. Typical design flaws include redundant constraints or inconsistencies arising from unexpected interactions among integrity constraints. These errors are far from trivial and may be very hard to detect and diagnose at the code level.

In this context, we have developed two open source tools capable of analyzing UML/OCL class diagrams: UMLtoCSP and its evolution, EMFtoCSP, which is able to deal with more general EMF-based models and is integrated within the Eclipse IDE. These tools frame correctness properties as Constraint Satisfaction Problems (CSPs), whose solution is an instance that constitutes an example that proves (or a counterexample that disproves) the property being checked. This mapping is transparent to the user, which means that they do not need a formal methods



(a)

OCL constraints selection

Select the file with the model OCL constraints

- ER
  - .project
  - models
    - ER.ecore
    - ER.ecore.ocl
    - ER.ecorediag
  - VerificationResults
- eShop
- PaperResearchers

□ Skip selection

(b)

Domains and cardinalities

Set the attributes domain and the cardinalities of the classes and associations

Model Elements	Value
Classes	
Schema	0..1
Entity	0..4
Relship	0..4
name: BigInteger	1..50
RelshipEnd	0..6
Attribute	0..5
Associations	
entities_schema	0..10
relships_schema	0..10

(c)

Properties selection

Select the properties to be checked

Model properties
☑ Strong Satisfiability
□ Weak Satisfiability
□ Liveliness: Schema

Constraints properties
□ Lack Of Constraints Subsumptions: ERN, ERN
□ Lack Of Constraints Redundancies: ERN, ERN

(d)

Validation results

Select the folder where the validation results will be stored

- ER
  - .project
  - models
  - VerificationResults
- eShop
- PaperResearchers

(e)

Figure 2: The EMFtoCSP interface, including the selection of the constraint (a), bounds (b) and properties (c) used to make the verification. The visualization of the results are presented in (d) and (e).

background to execute the tools or understand their output.

The constraint-logic programming solver ECLiPSe is used as the reasoning engine to find instances. This choice offers advantages and disadvantages with respect to SAT solvers, e.g., better support for complex numerical constraints. As with all bounded verification approaches, when a solution is not found, no conclusion can be drawn about the property since a solution could exist beyond the search bounds.

We believe that using this approach could enable the adoption of model verification practices in many software companies, where currently none are employed due to the lack of a usable solution. Ultimately, this absence risks the quality of the software they produce. Nevertheless, there is still a lot of work to be done. A number of extensions are currently planned for EMFtoCSP, but we would like to highlight two. The first is incremental verification, where once a model has been checked, further evaluation should only consider the subset of the model that has changed since the last run. The second is an automatic suggestion of search bounds whereby a quick pre-analysis of the model could suggest promising bounds within which we should look for a solution to maximize the performance.

Link:
EMFtoCSP:
https://github.com/atlanmod/EMFtoCSP

Reference:

[1] C. Jones, P. O'Hearn, J. Woodcock: "Verified Software: A Grand Challenge", Computer, vol. 39, no. 4, pp. 93-95, April 2006.

Please contact:

Jordi Cabot Sagrera
Inria and École des Mines de Nantes, France
E-mail: jordi.cabot@inria.fr

Robert Clarisó Viladrosa
Internet Interdisciplinary Institute – Universitat Oberta de Catalunya
E-mail: rclariso@uoc.edu

KandISTI: A Family of Model Checkers for the Analysis of Software Designs

by Maurice ter Beek, Stefania Gnesi and Franco Mazzanti

Driven by a series of European projects, researchers from the Formal Methods and Tools lab of ISTI-CNR have developed a family of model-checking tools for the computer-aided verification of the correctness of software designs. To date, these tools have been applied to a range of case studies in the railway, automotive and telecommunication fields.

When analyzing the correctness of designs in complex software systems during their early stages of development, it is essential to apply formal methods and tools. The broader system is described using a formal specification language and its relative correctness (with respect to relevant behavioural properties) is checked by formally evaluating temporal logic formulas over the underlying computational model. Over the last two decades, we have developed the KandISTI family of model checkers, each one based on a different specification language, but all sharing a common (on-the-fly) temporal logic and verification engine.

The main objective of the KandISTI framework is to provide formal support to the software design process, especially in the early stages of the incremental design phase (i.e., when designs are still likely to be incomplete and likely to contain mistakes). The main features of KandISTI focus on the possibilities of (i) manually exploring the evolution of a system and generating a summary of its behaviours; (ii) investigating abstract system properties using a temporal logic supported by an on-the-fly model checker; and (iii) obtaining a clear explanation of the model-checking results, in terms of possible evolutions of the specific computational model.

The first tool in the family was the FMC model checker which described a system by a hierarchical composition of sequential automata. This tool proved to be a very useful aid when teaching the fundamentals of automated verification techniques in the context of software engineering courses. As an attempt to reduce the gap between theoreticians and software engineers, the original model-checking approach was experimented over a computational model based on UML statecharts. In the context of the FP5 and FP6 EU projects AGILE and SENSORIA, this has led to the development of UMC, in which a system is specified as a set of communicating UML-like state machines.

In cooperation with Telecom Italia, UMC was used to model and verify an asynchronous version of the SOAP communication protocol and model and analyse an automotive scenario provided by an industrial partner of the SENSORIA project. Currently UMC is being used successfully in the experimentation of a model-checking-based design methodology in the context of the regional project TRACE-IT (Train Control Enhancement via Information Technology). This project aims to develop an automatic train supervision system that guarantees a deadlock-free status for train dispatches, even when there are arbitrary delays with respect to the original timetable. The largest model we analysed in this context had a statespace of 35 million states.

Again in the context of SENSORIA, we developed the CMC model checker for the service-oriented process algebra COWS. Service-oriented systems require a logic that expresses the correlation between dynamically generated values appearing inside actions at different times. These values represent the correlation values which allow, e.g., to relate the responses of a service to their specific requests or to handle the concept of a session involving a long sequence of interactions among interacting partners. CMC was used to model and analyse service-oriented scenarios from the automotive and finance domains, as provided by industrial partners in the project.

The most recent member of the KandISTI family is VMC, which was developed specifically for the specification and verification of software product families. VMC performs two kinds of behavioural variability analyses on a given family of products. The first is a logic property expressed in a variability-aware version of a known logic. This can directly be verified against the high-level specification of the product family behaviour, relying on the fact that under certain syntactic conditions the validity of the property over the family model guarantees the validity of the same property for all product models of the family. The second is that the actual set of

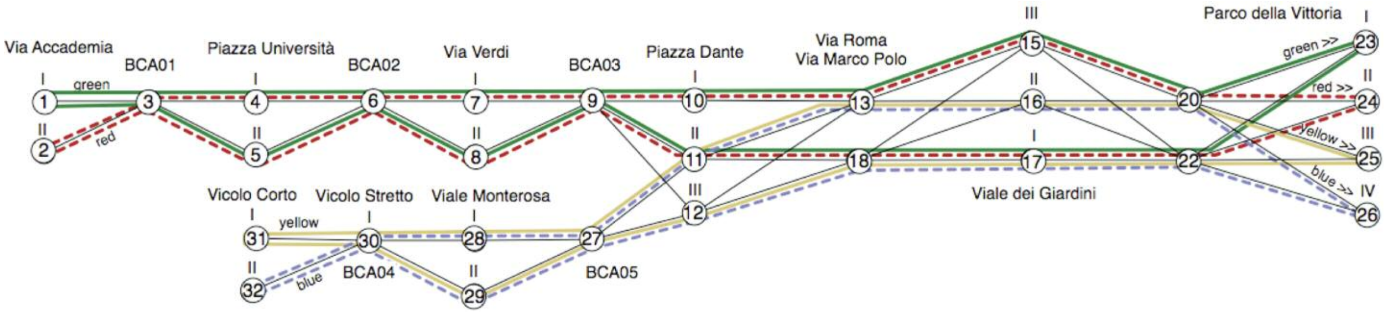


Figure 1: The railway yard layout and missions for trains on the green, red, yellow and blue lines.

valid product behaviours can be generated explicitly and the resulting specifications can be verified against the same logic property (this is surely less efficient than direct verification, but it makes it possible to identify precisely why the original property failed over the whole family). Experimentation with VMC is on-going in the context of the EU FP7 project QUANTICOL. To date, only a small version of the bike-sharing case study from QUANTICOL has been considered but more effort is needed to evaluate VMC on more realistically sized problems.

Link: http://fmt.isti.cnr.it/kandisti

References:
[1] M.H. ter Beek, A. Fantechi, S. Gnesi, and F. Mazzanti. “A state/event-based model-checking approach for the analysis of abstract system properties.” Science of Computer Programming 76(2):119-135, 2011
http://dx.doi.org/10.1016/j.scico.2010.07.002
[2] A. Fantechi, S. Gnesi, A. Lapadula, F. Mazzanti, R. Pugliese, and F. Tiezzi. “A logical verification methodology for service-oriented computing.” ACM Transaction on Software Engineering and Methodology 21(3):16, 2012.
http://doi.acm.org/10.1145/2211616.2211619
[3] F. Mazzanti, G. O. Spagnolo, and A. Ferrari. “Designing a deadlock-free train scheduler: A model checking approach.” NASA Formal Methods, LNCS 8430, Springer, 2014, 264-269.
http://dx.doi.org/10.1007/978-3-319-06200-6_22

Please contact:
Franco Mazzanti, ISTI-CNR, Italy
E-mail: franco.mazzanti@isti.cnr.it

QVTo Model Transformations: Assessing and Improving their Quality

by Christine M. Gerpheide, Ramon R.H. Schiffelers and Alexander Serebrenik

Applying a unique bottom-up approach, we (Eindhoven University of Technology (The Netherlands)) worked collaboratively with ASML, the leading provider of complex lithography systems for the semiconductor industry, to assess the quality of QVTo model transformations. This approach combines sound qualitative methods with solid engineering to proactively improve QVTo quality in a practitioner setting.

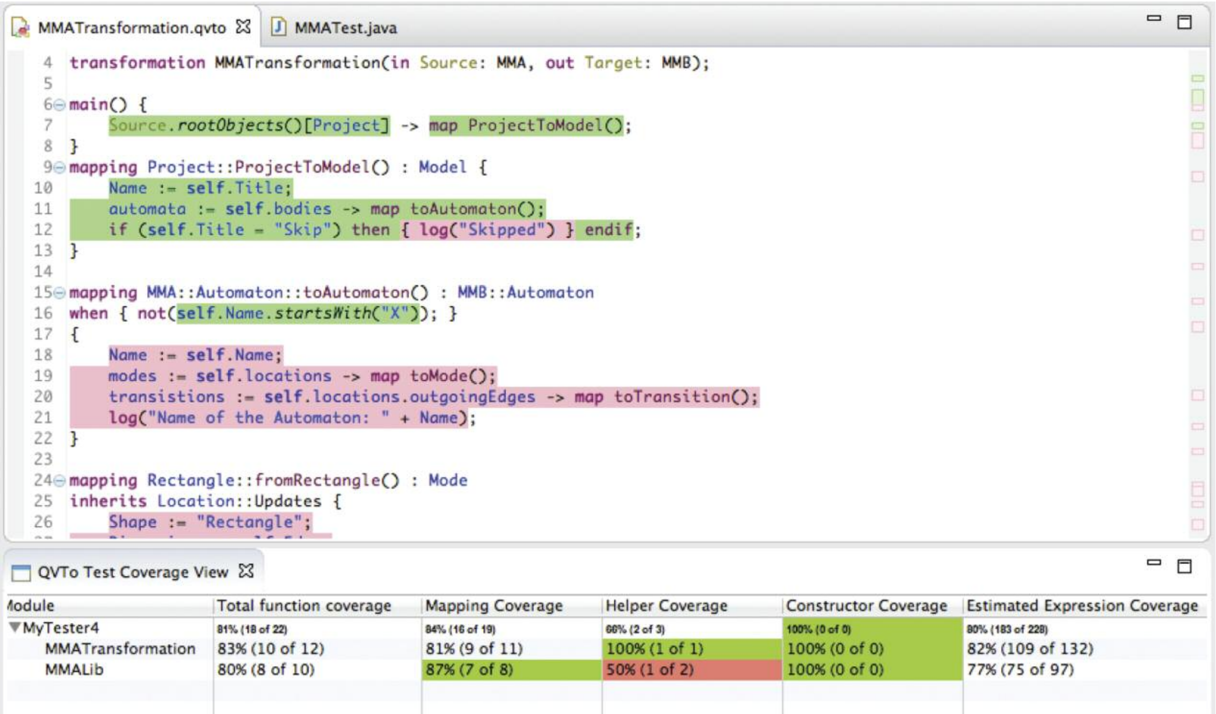
Model-driven engineering (MDE) can be used to develop highly reliable software which offers a range of benefits from systems analysis and verification to code generation. In MDE, system models are created by domain experts and then transformed into other models or code using model transformations. One of the languages used for writing these model transformations is QVT Operational Mappings (QVTo) which was specified in the 2007 Object Management Group (OMG) standard for model-to-model transformation languages. QVTo is regularly used by both academics and industry practitioners, including ASML, the leading provider of complex lithography systems for the semiconductor industry. Currently ASML has more than 20,000 lines of QVTo code, supporting more than a hundred model transformations.

Despite its widespread use, however, QVTo is a relatively new language. For general-purpose languages, developers have had time to share best practices and establish a number of standard reference points against which the quality of a piece of code can be judged. These are yet to be developed for QVTo. Moreover, QVTo has a large amount of language-specific constructs which are not available in general-purpose languages or even in other model-to-model transformation languages. In fact, QVTo specifications have been described by some as “rather voluminous” and even “fantastically complex”. Therefore, it is unclear whether established intuitions about code quality apply to QVTo and a lack of standardized and codified best practices is recognized as a serious challenge in assessing their transformation quality.

In a response to this challenge, ASML and Eindhoven University of Technology have joined in an ongoing collaboration to investigate the quality of QVTo transformations. In addition to assessing the quality of a transformation, this project is also seeking to promote the creation of higher-quality transformations from their inception, improving quality proactively [1].

To achieve this goal, a bottom-up approach was used which combined three qualitative methodologies. To begin, a broad exploratory study which included the analysis of interviews with QVTo experts, a review of the existing literature and other materials, and introspection were completed. Then, a QVTo quality model was developed to formalize QVTo transformation quality: this model consisted of high-level quality goals, quality properties, and evaluation procedures. The quality model was validated using the outcomes from a survey of a broad group of QVTo developers in which they were asked to rate each model property on its importance to QVTo code quality.

Many of the model properties recognized as important for QVTo transformation quality are similar to those in traditional languages (e.g., small function size). However, this analysis also highlighted a number of properties which are specific to QVTo or other model transformation languages. For instance, we found that the following QVTo-specific properties were considered important for quality: the use of only a few black boxes, few queries with side effects, little imperative programming (e.g., for-loops) and small init sections. Deletion using trash-bin patterns was also found to be beneficial for per-



```
transformation MMATransformation(in Source: MMA, out Target: MMB);

main() {
    Source.rootObjects()[Project] -> map ProjectToModel();
}
mapping Project::ProjectToModel() : Model {
    Name := self.Title;
    automata := self.bodies -> map toAutomaton();
    if (self.Title = "Skip") then { log("Skipped") } endif;
}

mapping MMA::Automaton::toAutomaton() : MMB::Automaton
when { not(self.Name.startsWith("X")); }
{
    Name := self.Name;
    modes := self.locations -> map toMode();
    transistions := self.locations.outgoingEdges -> map toTransition();
    log("Name of the Automaton: " + Name);
}

mapping Rectangle::fromRectangle() : Mode
inherits Location::Updates {
    Shape := "Rectangle";
```

Module	Total function coverage	Mapping Coverage	Helper Coverage	Constructor Coverage	Estimated Expression Coverage
MyTester4	81% (18 of 22)	84% (16 of 19)	66% (2 of 3)	100% (0 of 0)	80% (183 of 228)
MMATransformation	83% (10 of 12)	81% (9 of 11)	100% (1 of 1)	100% (0 of 0)	82% (109 of 132)
MMALib	80% (8 of 10)	87% (7 of 8)	50% (1 of 2)	100% (0 of 0)	77% (75 of 97)

Figure 1: The coverage tool highlights the covered and not covered parts of the QVTo transformation as well as reports coverage-related statistics

formance as it has high test coverage in relation to functionality [2].

Test coverage also emerged as a key issue in the expert interviews, with every interviewee highlighting the lack of a test coverage tool. Consequently, we prioritized this demand and created a test coverage tool for QVTo (Figure 1). Implemented as an Eclipse plugin, the tool helps QVTo developers to create higher-quality transformations from the start: at the developers' request, the tool reports coverage percentages for different units of QVTo transformations (e.g., mappings, helpers, constructors), as well as visualizing expressions covered and not covered by the tests. During the seven week study period, the tool was used 98 times, resulting in execution of 16,714 unit tests. In addition to assisting with debugging issues, the tool was also used by one developer to prepare a user story, a description of how to complete the specific task of an agile sprint. When preparing the user story, the developer ran the entire test suite and looked at the coverage for the modules he knew would be affected by the task to be completed. He then noted exactly which places in the modules were not covered by the test suite by inspecting the coverage overlay. Then, as part of the list of steps for the feature, he added an additional step stating that before implementation of the feature can begin, additional tests must be written to cover the untested parts. The developers also stated the tool easily identifies dead code. The coverage tool has also been added to the development team's "Way of working" document, making it an official part of their development process.

As a side product of our research, three patches were submitted, accepted and integrated into the Eclipse QVTo core engine which was released with Eclipse Luna on June 25, 2014. Together, these patches make it possible for QVTo interpreters to easily access the test coverage tool, as well as future tools, such as an integrated profiler [3].

References:

[1] C.M. Gerpheide: "Assessing and Improving Quality in QVTo Model Transformations", M.Sc. thesis, Eindhoven University of Technology, 2014, http://phei.de/docs/academic/Gerpheide_QualityQVToTransformations.pdf

[2] C.M. Gerpheide, R.R.H. Schiffelers, A. Serebrenik: "A Bottom-Up Quality Model for QVTo", QUATIC 2014, Portugal, 2014)

[3] C.M. Gerpheide et al.: "Add hooks for contributing 3rd-party visitor decorators", https://bugs.eclipse.org/bugs/show_bug.cgi?id=432969

Please contact:
Alexander Serebrenik
Eindhoven University of Technology, The Netherlands
Tel: +31402473595
E-mail: a.serebrenik@tue.nl

Redundancy in the Software Design Process is Essential for Designing Correct Software

by Mark G.J. van den Brand and Jan Friso Groote

Researchers at Eindhoven University of Technology in the Netherlands plead the case for more redundancy in software development as a way of improving the quality of outcomes and reducing overall costs.

If an engineer was asked how reliable a critical artefact must be, a likely answer might be that a system can only fail once in 10^{10} times. For example, the probability of a ship's hull collapsing during its lifetime or an active high water barrier failing to function when it is requested to do so, are typically in the order of 10^{-10}. These numbers are so low that no engineer will ever experience the failure of his own artefact.

Considering such a question led us to reflect on how it may be possible to obtain similar numbers when designing software. In our previous work, we addressed this question by constructing a simple failure model and, we found a simple answer [1]. The only way to obtain such figures was by employing redundancy. Such an approach is very common in the more classical engineering disciplines. For example, when considering support columns with a failure probability of 10^{-5}, an engineer can simply use two columns (where only one is truly necessary), thus allowing the overall failure probability to be increased to 10^{-10}.

To examine how this redundancy approach applies in the software development field, we must realise that software is very different from physical artefacts. In physical artefacts, components fail due to wear and tear while software fails due to built-in errors, for example, a small typing error, an incorrect algorithm or the wrong use of an interface. When such an error is activated, the software fails. As software has many varied states, it can take a long time for some errors to become active, although as shown in [2] many programming faults lead to a multitude of erroneous states. Therefore, these latter faults are far easier to catch.

The probability of a hardware failure is almost negligible and thus, can be ignored. Software errors are always directly caused by either the programmers or program designers that left those errors in the code. As humans they have a large probability of doing something wrong [3]. At best, the failure probability of humans is only 10^{-3} but even this figure can only be applied in situations where the tasks are very simple and the programmer highly trained. For more complex tasks, failure probabilities of 10^{-2} or 10^{-1} are more realistic. In situations where a human must complete a non-trivial task under stress, they are almost certain to fail.

This footbridge (Nederwetten, The Netherlands) is made of more steel than strictly necessary to assure its quality. Software engineers must also consciously employ redundancy to ensure quality.

It should be obvious that the difference between the failure probability of a programmer and the desired failure probability of a critical piece of software is around eight orders of magnitude. Obvious measures, such as comprehensive training for programmers or the use of the most modern programming languages are excellent solutions but alone, these measures are unable to bridge this gap. Training can never accomplish an improvement of more than a factor 10^{-2} and for a complex task such as programming, even this is unlikely. Using modern programming languages, even domain specific languages, in combination with libraries can lead to substantial reductions in the amount of required code and thus, reduce the overall numbers of errors. However, here too, the possible reductions that can be achieved (at most a factor of 100) are insufficient.

Thus, the only way to achieve the desired failure probability of 10^{-10} is to consciously employ redundancy in the software design process. Typically, when constructing software, it must be described in several ways. These differing approaches should then be meticulously compared and challenged, with the goal of removing as many of the flaws that will be inherent in each description.

Several forms of redundancy are already present in actual programming, such as type checking and testing. However, these forms of redundancy came about as good practices, not conscious ways to introduce redundancy with a view to attaining a certain level of software quality.

Active redundancy can be brought into the software design process through the introduction of high level models of the software, for instance, in the form of domain specific languages, property languages such as modal logics to independently state properties, independently (and perhaps multiple) constructed implementations, and a priori described test cases. The comparison of these different views can be done by model checking (software or models against properties), model based testing (model against implementation), and systematic testing (tests against model or software). Code inspection and acceptance tests are also fruitful, but lack the rigour of comparison that the more mathematical methods have.

By acknowledging that redundancy in design is the only way to obtain reliable software, one can then question certain trends. For instance, there is an ongoing trend to eliminate the annoyance associated with static type checking. A language like Python is a typical example, as is the introduction of the auto keyword in C++ which allows a programmer to skip writing down an explicit type. The desire for code generation out of a model, or research in generating a model and software out of requirements put on the software introduce a single point of failure in the design process. These approaches do not pay tribute to the need for redundancy and discourage the detection of flaws that are inevitably made in the design of the software. This is a serious problem, as we all know that such flaws can wreak havoc when they happen to be activated.

Links:
http://www.win.tue.nl/~mvdbrand/
http://www.win.tue.nl/~jfg/

References:

[1] M.G.J. van den Brand and J.F. Groote: " Software Engineering: Redundancy is Key", J.J. Vinju editor, preprint Science of Computer programming, Special issue in Honour of Paul Klint, pp. 75-82, 2013.

[2] J.F. Groote, R. van der Hofstad and M. Raffelsieper: "On the Random Structure of Behavioural Transition Systems", Computing Science Report CS-R1401, Department of Mathematics and Computer Science, Eindhoven University, 2014. http://www.win.tue.nl/~jfg/articles/CSR-14-01.pdf

[3] D.J. Smith: "Reliability, maintainability and risk. Practical Methods for Engineers", Elsevier, 2011.

Please contact:
Mark G.J. van den Brand
Eindhoven University of Technology, The Netherlands
Tel: +31402472744
E-mail M.G.J.v.d.Brand@tue.nl

Jan Friso
Eindhoven University of Technology, The Netherlands
E-mail J.F.Groote@tue.nl

Estimating the Costs of Poor Quality Software: the ICEBERG Project

by Luis Fernández, Pasqualina Potena and Daniele Rosso

Project ICEBERG investigated a novel approach to improving understanding of the real cost impacts of poor quality software and supporting the suite of management decisions required to take corrective action across the entire software development cycle.

The ICEBERG project was developed to consider the issue of Transfer of Knowledge (ToK) in the Software Quality Assurance (QA) domain and had two main objectives: (1) investigating, defining and implementing model-based processes oriented to identifying the most effective and efficient QA strategy for software development in general, and more specifically, software developed for telecommunications and finance organisations; and, as stated for this type of Marie Curie projects, (2) bolstering the research platform in this area for future work through the secondment of researchers and the specific training of early stage and recruited researchers.

Project Motivation

Commonly, software projects need to be performed and delivered against project schedules that specify timings, costs and quality constraints (amongst other things). One of the most cost- and time-intensive components of the overall development cycle is the QA process. A major issue associated with this process is that the individual analysis of single factors in isolation is frequently inaccurate, as pairs of factors may visibly (and sometimes adversely) affect each other. Therefore, frameworks that support decisions made in relation to meeting scheduling and quality requirements, while keeping project costs within budget, would be very helpful for project managers.

Research Themes and Challenges

The ICEBERG project started in February 2013 and will end in December 2017. It is funded through the European Marie Curie program (IAPP category). The project's main scope is to provide researchers with new research skills and broaden the horizons of models-based processes with a view to identifying the most effective and efficient QA strategy in software development.

A number of institutions collaborated on this project: two research centres (CINI (Consorzio Interuniversitario Nazionale per l'Informatica) - University of Naples and University of Alcalá (UAH)) and two SMEs (Assioma.net and DEISER). Specifically, the two universities provided skills in the areas of quality estimation and forecasting models of software products/processes and related costs. The SMEs contributed highly qualified real-life experience on the testing of software projects/processes. The project involves up to 19 researchers who all have the opportunity to make cross-entity swaps with the other partner institutions. The researchers then have the opportunity to share their capacities, acquire new skills and develop new competences on decision support systems in the quality assurance domain. Once they return, this knowledge flow continues, this time back to their home institutions, enhancing European economic and scientific competitiveness. Up to three researchers have been specifically contracted for periods of 18 or 24 months in order to contribute to the project and to be trained as specialists in the field.

The key focus of the project will be the enhanced support that a joint analysis of schedules/times, costs and quality can

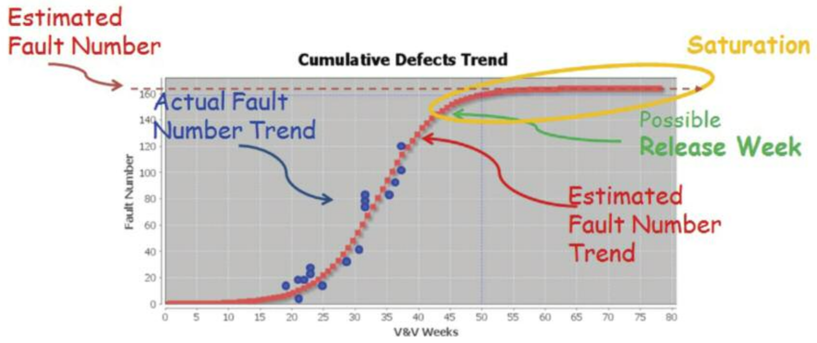


Figure 1: A graphical illustration of the decision model indicating when it is best to stop testing and move into the software delivery phase [2].

give to decision-making (details on the project scope can be found in [1]). A particular emphasis will be given to the design and development of innovative and effective models for (1) evaluating the costs associated with testing activities in relation to a given quality issue (e.g., missing, incomplete or wrong implementation of testing activities/phases) and (2) guiding business decision processes on what investments should be made in the software testing process (e.g., see Figure 1).

Longer-term, the objectives of the ICEBERG project include: (1) the creation of a database which enables data collected from the literature and past business (software) projects (provided by industrial partners) to be categorised; (2) the definition of model-based processes to support decision-making on investment in testing activities [3] (e.g., the scheduling and allocation of various testing activities and the effort in each phase defined in the test plan); and (3) the development of proof-of concept IT tools for automating the application of model-based processes.

Both the model-based processes and proof-of concept IT tools will be evaluated using real-world test cases provided by the industrial partners. It will be the first attempt to combine existing literature and practical experience (provided by experts in the field). The decision-making frameworks developed through this project will help to maximize the effectiveness of practitioners. The adoption of well-assessed quality decision methods can only be effectively achieved by analyzing the effort and time necessary to incorporate them into real-world systems. Therefore, we know that understanding practitioners perceptions regarding the strengths, limitations and needs associated with using state of the art practice solutions in industry is vital. We hope that once completed, the outcomes of this work will address the classical questions "How many tests are enough?" and "When to stop software testing?"

Link:
ICEBERG: http://www.iceberg-sqa.eu/

References:

[1] P. Potena et al.: "Creating a Framework for Quality Decisions in Software Projects", ICCSA (5) 2014: 434-448, LNCS.
[2] M. Cinque et al.: "On the Impact of Debugging on Software Reliability Growth Analysis: A Case Study", ICCSA (5) 2014: 461-475, LNCS.
[3] L. Fernandez, P. J. Lara, J. J. Cuadrado: "Efficient Software Quality Assurance Approaches Oriented to UML Models in Real Life", Verification, Validation and Testing in Software Engineering, IGI Global, 385-426, 2007.

Please contact:

Luis Fernández, Pasqualina Potena
University of Alcalá, Spain,
E-mail: luis.fernandez@uah.es,
p.potena@uah.es

Daniele Rosso
Assioma.net srl, Italy
Email: daniele.rosso@assioma.net

Software Quality in an Increasingly Agile World

by Benoît Vanderose, Hajer Ayed and Naji Habra

For decades, software researchers have been chasing the goal of software quality through the development of rigorous and objective measurement frameworks and quality standards. However, in an increasingly agile world, this quest for a perfectly accurate and objective quantitative evaluation of software quality appears overrated and counter-productive in practice. In this context, we are investigating two complementary approaches to evaluating software quality, both of which are designed to take agility into account.

Most software engineering processes and tools claim to assess and improve the quality of software in some way. However, depending on its focus, each one characterizes quality and interprets evaluation metrics differently. These differences have led to software researchers questioning how quality is perceived across various domains and concluded that it was an elusive and multifaceted concept. However, two key perspectives stand out: a software's objective quality and subjective quality.

The objective "rationalized" perspective is taught in influential quality models and promoted through standards such as ISO/IEC 25010. It envisions quality as conformance to a pre-defined set of characteristics: quality is an intrinsic data of the product that can be measured and must be compared against a standard in order to determine its relative quality level. Therefore, from this perspective, quality assurance is closely associated with quality control.

The subjective perspective, on the other hand, defines software quality as a constantly moving target based on customer's actual experiences with the product. Therefore, quality has to be defined dynamically in collaboration with customers as opposed to predefined standards. This definition welcomes change that enhance the quality of the customer's experience, emphasizes the possibility to tolerate deliberately bad quality (in order to do it better next time) and allows quality goals to be redefined. As such, quality is constructed and checked iteratively and can evolve over time. This then leads to *constructive quality* or *emergent quality*. The subjective perspective also promotes the idea of on-going customer's satisfaction and garners everyone's commitment to achieving quality: thus, quality assurance becomes an organization-wide effort or what is called *holistic quality*.

The suitability of either perspective depends on the software development process. In a production context, quality is defined as the conformance to set requirements while quality in a service context should take into account the fact that each stakeholder will have a different definition of what constitutes a quality experience, and furthermore, these perceptions will evolve over time. In the field of software engineering, there has been a move from the compliance view towards a constructive holistic quality assurance view. This is particularly notable in the case of iterative and incremental software engineering methods and agile methods.

Improving the support for this way of envisioning software quality is one of the research topics addressed by the PReCISE research center at the University of Namur, and current efforts focus on two complementary research areas: model-driven quality assessment and iterative context-driven process evolution.

MoCQA and AM-QuICk frameworks

Our attempts to capture the essence of a "traditional" quality assessment (i.e., metrics, quality models and standards) in a unified meta-model resulted in a fully-fledged model-driven quality assessment framework named MoCQA [1]. This framework aims to provide the methodology, tools and guidelines to integrate evaluation methods from different sources and associate them with a quality goal, a set of stakeholders and an artefact (e.g., a piece of code, UML diagram, etc.), allowing these elements to coexist in a coherent way. Being model-driven, the framework provides a unified view of the quality concerns of specific stakeholders and iteratively guides the actual assessment. However, in order to leverage the benefits of the framework, it is essential to perform the quality assessment iteratively and incrementally (the feedback from the assessment helps improve the products) and ensure that this feedback is taken into account to pilot the next steps of the development process.

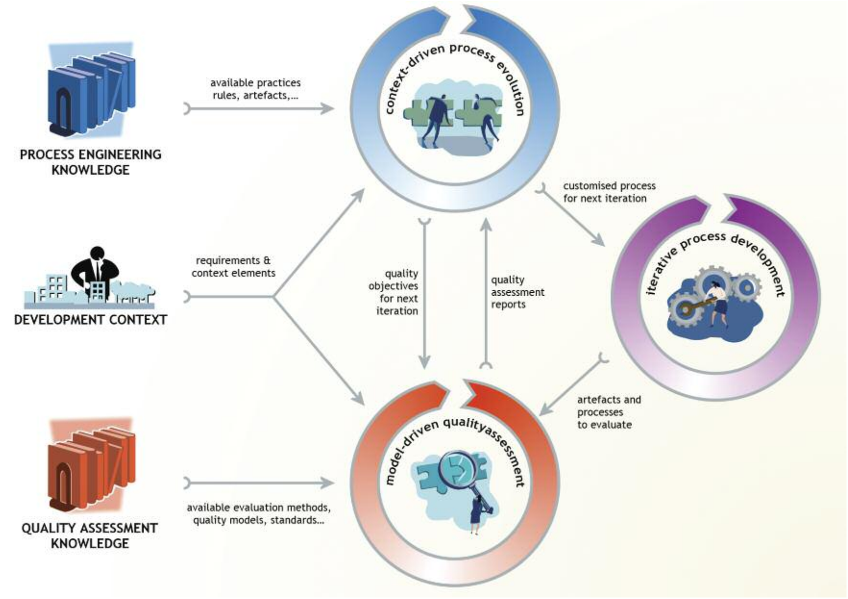

PROCESS ENGINEERING KNOWLEDGE
available practices rules, artefacts,...
context-driven process evolution
DEVELOPMENT CONTEXT
requirements & context elements
customised process for next iteration
quality objectives for next iteration
quality assessment reports
iterative process development
artefacts and processes to evaluate
QUALITY ASSESSMENT KNOWLEDGE
available evaluation methods, quality models, standards...
model-driven qualityassessment

Figure 1: An agile and quality-oriented development process based on the complementarity between the model-driven quality assessment (MoCQA) and the iterative context-driven process evolution (AM-QuICk).

Guaranteeing the positive impact of an assessment on the development process calls for iterative process evolution. Our research in the field of agile methods customization [2] revealed that this customization does not include the fact that the context itself may evolve over time. Another framework, AM-QuICk [3] is designed to allow a truly context-driven evolution of the development process, a review at each iteration ensuring that it can be adapted to the current context. It relies on the elaboration of a repository of reusable agile artefacts, practices and metrics and a context-sensitive composition system.

In order to exploit the benefits of an iterative process evolution, decision-making elements are needed to guide the evolution and decide which practices to include at the right time. This can be achieved through model-driven quality assessment, making the two approaches complementary (Figure 1).

Future Work

Looking to the future, our efforts will focus on tightening the integration between the two frameworks. Advancements in these complementary research areas offer great opportunities to provide development teams with comprehensive sets of tools with which to manage an evolving software development process that focuses on the global satisfaction of each stakeholder at each stage. Such tools ensuring these processes can operate more effectively in an increasingly agile world.

References:

[1] B. Vanderose, "Supporting a model-driven and iterative quality assessment methodology: The MoCQA framework," Ph.D. dissertation, Namur Univ., Belgium, 2012

[2] H. Ayed et al., "A metamodel-based approach for customizing and assessing agile methods," in proc. 2012 QUATIC.

[3] H. Ayed et al., "AM-QuICk : a measurement-based framework for agile methods customization," in proc. 2013 IWSM/MENSURA.

Please contact:

Naji Habra, Hajer Ayed or Benoît Vanderose
University of Namur, Belgium
naji.habra@unamur.be
hajer.ayed@unamur.be
benoit.vanderose@unamur.be

Improving Small-to-Medium sized Enterprise Maturity in Software Development through the Use of ISO 29110

by Jean-Christophe Deprez, Christophe Ponsard and Dimitri Durieux

Technological small to medium sized enterprises (SMEs) that are involved in developing software often lack maturity in the development process. This can result in an increased risk of software quality issues and technological debt. Recently, standards have started to address the need for lightweight process improvements through the development of the ISO 29110 series. In this article, we look at how this series can successfully be applied in the areas of (self)-assessment, diagnosis and specific tooling support.

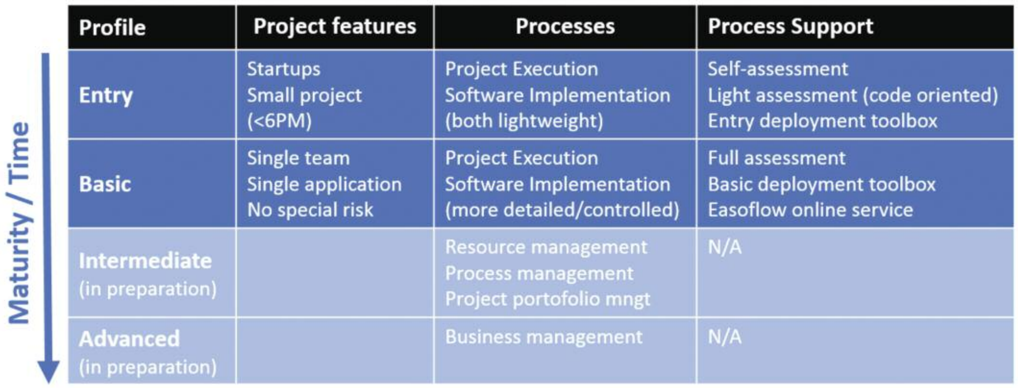
SMEs and the small IT sections of larger companies often produce valuable software products. Due to limited resources, coping with quality issues in these very small entities (VSEs) is not an easy task. In these contexts, using process reference models such as CMMI or ISO 12207 are clearly excessive. Therefore, the need for a lightweight standard is useful and will particularly assist VSEs that need recognition as suppliers of high quality systems. Such a standard would also provide a practical process improvement tool.

Through the CE-IQS project, CETIC has been promoting the use and development of such lightweight methods for years. The group has shared its experiences with the working group 24 (WG24) of sub-committee 7 (SC7) of the Joint Technical Committee 1 (JTC1) of the International Organization for Standardization and the International Electro-technical Commission (ISO/IEC). This work has brought together contributions from numerous parties including Canada (ETS), Ireland (LERO), Japan, Thailand and Brazil, and has resulted in a new ISO/IEC 29110 series of standards. These new standards address the “Systems and Software Life Cycle Profiles and Guidelines for Very Small Entities” [1] and have been made freely available by the ISO [2]. Of particular relevance is Part 5 which contains dedicated guidelines for VSEs that are complemented by deployment packages to assist with their adoption.

The ISO/IEC 29110 series are structured to follow a progressive approach which is based on a set of profiles that range from an entry profile to the most advanced profile (see Figure 1). The entry profile consists of the simplest set of development practices, covering software implementation and project management activities, and is particularly suited to start-ups. This is followed by the basic, intermediate and advance profiles that progressively cover a growing set of activities that can handle more complex situations involving a larger range of risks. The intermediate profile, although progressing well, is yet to be published and the advanced profile is still being discussed by the WG24.

In addition to contributing to this work, CETIC is also actively applying the standard as a practical tool for increasing the maturity of VSEs developing software. This work is not being done in a certification context as there is little incentive for this measure in Belgium as opposed to Thailand and Brazil where VSEs are highly involved in off-shoring (and hence, want to obtain such a certification to advertise the quality of their work). Instead, our actions take the following forms:

- *Self-assessment questionnaire:* through a dedicated web-site, companies can answer a set of questions to see how their current development approach compares against the entry profile activites. The questionnaire is currently available in French, English and Czech and feedback and/or translations into other languages are welcomed.
- *Light assessment:* a two-hour assessment which is based on a two-page check-list based on the entry profile. This assessment can be conducted as part of a code quality assessment in a VSE context (has become mandatory to integrate some start-up incubators in Wallonia).
- *Full assessment:* comprises of one to two days of interviews with key personnel including the project manager, architect and possibly some developers. The assessment should be conducted in two phases so that weak


Maturity / Time

Profile	Project features	Processes	Process Support
Entry	Startups Small project (<6PM)	Project Execution Software Implementation (both lightweight)	Self-assessment Light assessment (code oriented) Entry deployment toolbox
Basic	Single team Single application No special risk	Project Execution Software Implementation (more detailed/controlled)	Full assessment Basic deployment toolbox Easoflow online service
Intermediate (in preparation)		Resource management Process management Project portofolio mngt	N/A
Advanced (in preparation)		Business management	N/A

Figure 1: ISO29110 generic profile group and related process support

points can be both identified and further investigated, allowing a full set of recommendations to be produced. Additional coaching can also be provided.
- *Process support service:* an online service called "Easoflow" is currently being developed which will aim to help companies standardize their project management and software development processes by providing a simple and guided streamlining of the ISO/IEC 29110 entry and basic profile activities.

The results of all these activities are reported back to the ISO. Our on-going R&D work will focus on extending the standard, with specific profiles for areas such as system engineering, the development of Easoflow and the associated lightweight risk assessment tools.

Links:
CE-IQS project :
https://www.cetic.be/CE-IQS,1191
Deployment Packages:
http://profs.etsmtl.ca/claporte/English/VSE/index.html
Easoflow tool:
https://demo.easoflow.com
Self-assessment:
http://survey.cetic.be/iso29110/survey.php

References:
[1] C. Laporte et al.: "A Software Engineering Lifecycle Standard for Very Small Enterprises", in R.V. O'Connor et al. (Eds.): EuroSPI 2008
[2] ISO/IEC 29110:2011, Software Engineering - Lifecycle Profiles for Very Small Entities (VSEs), http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html

Please contact:
Christophe Ponsard
CETIC, Belgium
Tel: +32 71 490 743
E-mail: christophe.ponsard@cetic.be

Software Product Quality Evaluation Using ISO/IEC 25000

by Moisés Rodríguez and Mario Piattini

In recent years, software quality has begun to gain great importance, principally because of the key role software plays in our day-to-day lives. To control software quality, it is necessary to conduct evaluations of the software products themselves. The AQC Lab was established for this purpose and its core responsibility is evaluating the quality of software products against the ISO/IEC 25000 standard.

Although numerous certifications for software quality exist (e.g., ISO/IEC15504, CMMI, etc.), there is little evidence to suggest that compliance with any of these standards guarantees good software products. Critics have gone so far as to suggest that the only thing these standards guarantee is uniformity of output and thus, may actually lead to the production of bad products. Consequently, the idea that software evaluations should be based on direct evidence of product attributes is becoming more widespread. Therefore, a growing number of organizations are becoming concerned about the quality of the products that they develop and/or acquire, as well as the processes.

The ISO/IEC 25000 family of standards, known as SQuaRE (Software Product Quality Requirements and Evaluation), appears to meet this emerging need to assess product-related quality. The objective of ISO/IEC 25000 is to create a common framework within which to evaluate software product quality and this standard is now beginning to replace the previous ISO/IEC 9126 and ISO/IEC 14598 standards to become the cornerstone of this area of software engineering. ISO/IEC25000 is divided into several parts: we highlight ISO/IEC25040 [1] which defines the process of evaluating software product quality and ISO/IEC25010 [2] which determines the software product characteristics and sub-characteristics that can be evaluated.

As with other standards, ISO/IEC 25000 describes what to evaluate but does not specify how. In other words, it does not detail the thresholds for the evaluation metrics to be used, nor does it describe how to group these metrics in order to assign a quality value to a software product.

Improving this evaluation process has been the goal for AQC team (began by the Alarcos Research Group, University of Castilla-La Mancha) over the last few years. This has led to the creation of AQC Lab [3], the first laboratory accredited for ISO/ IEC17025 by ENAC (National Accreditation Entity) to assess the quality of software products using the ISO/IEC25000. The lab has also been recognised by the ILAC (International Laboratory Accreditation Cooperation). This accreditation confirms the technical competence of the laboratory and ensures the reliability of the evaluation results. The AQC Lab uses three main elements to conduct the quality evaluations. These are:
- The Assessment Process which directly adopts the activities of ISO/IEC 25040 and completes them with specific laboratory roles and instructions which have been developed internally. This process produces an evaluation report that shows the level of quality achieved by the product and any software aspects that may require improvement.
- The Quality Model which defines the characteristics and metrics needed to evaluate the software product. This model was developed through the MEDUSAS (Improvement and Evaluation of Usability, Security and Maintainability of Software (2009-2012)) research project, funded by MICINN/FEDER. Although the AQC Lab evaluates multiple features of the model presented in ISO/IEC 25010, the accreditation initially focuses on the characteristic of maintainability, principally because maintenance is one of the most expensive phases of the development lifecycle and maintain-

Properties / Subcharacteristics	Analyzability	Modularity	Modifiability	Reusability	Testability
Coding Rules Violation	X	X	X	X	X
Code Documentation	X			X	
Complexity	X		X		X
Coupling		X	X	X	X
Dependency Cycles		X	X	X	X
Cohesion	X	X			
Structuring Packages	X	X		X	X
Structuring Classes	X	X			X
Methods Size	X	X			
Duplicate Code	X		X		X

Table 1: Relationship between the sub-characteristics and quality properties in the quality model (completed in Step 2 of the evaluation process).

ability is one of the features most frequently requested by software clients, Many clients seek products that maintain themselves or can be maintained by a third party. This model required considerable effort to develop, eventually resulting in a set of quality properties that are measurable from the source code and related to the sub-characteristics of quality proposed in ISO/IEC25010 (Table 1).

- The Evaluation Environment which largely automates the evaluation tasks. This environment uses measurement tools that are applied in the software product to combine the values obtained, assign quality levels to the model sub-characteristics and characteristics and, eventually, present them in an easily accessible manner.

In addition, the Spanish Association for Standardization and Certification (AENOR) has created a software product quality certification based on ISO/ IEC25000. To perform a certification, AENOR reviews the assessment report issued by the accredited laboratory and makes a brief audit of the company and the product. If everything is correct, the company is then given a certificate of quality for its software product.

During the past year, several pilot projects have been run using this certification scheme. Participating companies (from Spain and Italy) have been the first to be assessed and future projects are planned for companies in Colombia, Mexico and Ecuador.

Links:
http://iso25000.com/
http://www.alarcosqualitycenter.com/index.php/software-quality-evaluation-lab

References:

[1] ISO, ISO/IEC 25040 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - Evaluation process. 2011: Ginebra, Suiza.

[2] ISO, ISO/IEC 25010, Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models. 2011: Ginebra, Suiza.

[3] Verdugo, J., M. Rodríguez, and M. Piattinni, Using Agile Methods to Implement a Laboratory for Software Product Quality Evaluation, in 15th International Conference on Agile Software Development (XP 2014). 2014: Roma (Italia).

Please contact:

Moisés Rodríguez
Alarcos Quality Center, Spain
E-mail:
Moises.rodriguez@alarcosqualitycenter.com

Mario Piattini
University of Castilla–La Mancha, Spain
E-mail: mario.piattini@uclm.es

High-Level Protocol Engineering without Performance Penalty for Multi-Core

by Farhad Arbab, Sung-Shik Jongmans and Frank de Boer

CWI's Reo-to-C compiler can generate code that outperforms hand-crafted code written by a competent programmer. As such, compared to conventional approaches to parallel programming, our approach has the potential to not only improve software quality but also performance.

In 2012, we described in ERCIM News a novel approach to programming interaction protocols among threads on multi-core platforms. This approach is based on the idea of using the graphical coordination language Reo, which has been subject to ongoing development by the Formal Methods group at CWI since the early 2000s, as a domain-specific language for the compositional specification of protocols. Since then, we have developed several code generation techniques and tools for Reo, including the Reo-to-Java and Reo-to-C compilers.

In terms of software engineering and software quality there are many advantages of using Reo. Reo [1] provides declarative high-level constructs for expressing interactions and thus, programmers can specify their protocols at a more suitable level of abstraction that can be achieved by using conventional languages (e.g., Java or C). These conventional languages provide only error-prone low-level synchronization primitives (e.g., locks, semaphores). Moreover, because Reo has a formal semantics, protocols expressed in Reo can be formally analyzed to improve

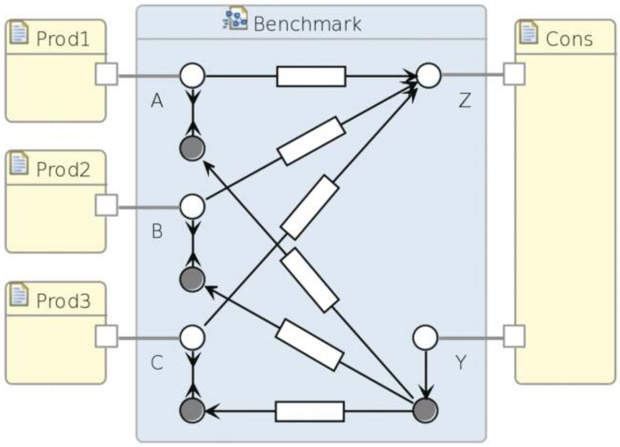


Figure 1: Producers-consumer protocol specification in Reo.

software quality and ensure correctness (e.g., by using model checking tools). Lastly, by using Reo, protocols become tangible, explicit software artifacts which promotes their reuse, composition and maintenance.

Two years ago, we thought that this list of software engineering advantages would be the main reason for programmers to adopt our approach, so long as the performance of the code generated by our compilers proved "acceptable". At the time, it seemed ambitious to strive for the level of performance (within an order of magnitude) that competent programmers can achieve hand-crafting code using a conventional language. After all, compiling high-level specifications into efficient lower-level implementations constitutes a significant challenge.

However, while developing our compilers, we came to realize that Reo's declarative constructs actually give us an edge, as compared to conventional languages. Reo allows for novel compiler optimization techniques that fundamentally conventional languages cannot apply. The reason for this is that Reo's declarative constructs preserve more of a programmer's intentions when they specify their protocols. When a sufficiently smart compiler knows exactly what a protocol is supposed to achieve, this compiler can subsequently choose the best lower-level implementation. Using conventional languages, in which programmers write such lower-level implementations by hand, information about their intentions is either irretrievably lost or very hard to extract. Therefore, to perform optimizations at the protocol logic level a compiler needs to reconstruct those intentions, but typically, it simply cannot. Thus, by using conventional languages for implementing protocols by hand, the burden of writing efficient protocol implementations rests exclusively on the shoulders of programmers, adding even more complexity to the already difficult task of writing parallel programs. As the number of cores per chip increases, the shortcomings of conventional programming languages in writing efficient protocol implementations will amplify this issue, effectively making such languages unsuitable for programming large-scale multi-core machines.

The following example, a simple producers-consumer program, offers the first evidence that our approach can result in better performance. In this program, every one of n producers produces and sends an infinite number of data elements to the consumer, while the consumer receives and consumes an infinite number of data elements from the producers. The protocol between the producers and the consumer states that the producers send their data elements asynchronously, reliably and in rounds. In every round, each producer sends one data element in an arbitrary order. A Reo specification realizes this protocol (for three producers; see Figure 1). We also had a competent programmer hand-craft a semantically equivalent program in C and Pthreads.

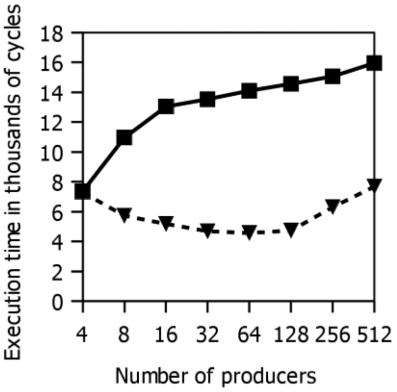


Figure 2: The results of a scalability comparison between the code generated by the Reo-to-C compiler (dashed line) and the hand-crafted implementation (continuous line).

We compared the scalability of the code generated by our current Reo-to-C compiler with the hand-crafted implementation (Figure 2). The generated C code runs on top of a novel runtime system for parallel programs. In this example, for this protocol, the Reo-based implementation outperformed the carefully hand-crafted code.

The technology behind our compiler is based on Reo's automaton semantics. Our most recent publication contains references to the relevant material [3]. All the optimization techniques used by the compiler have a strong mathematical foundation and we have formally proved their correctness (which guarantees correctness by construction).

We do not claim that code generated by our compiler will outperform every hand-crafted implementation in every protocol and in fact, know that it does not. However, we do believe that these first results are very encouraging and see several ways to further optimize our compiler in the future.

Links:

http://reo.project.cwi.nl
http://www.cwi.nl/~farhad

References:

[1] F. Arbab: "Puff, The Magic Protocol", in "Formal Modeling: Actors, Open Systems, Biological Systems (Talcott Festschrift)", LNCS 7000, pp. 169--206, 2011

[2] S.-S. Jongmans, S. Halle, F. Arbab: "Automata-based Optimization of Interaction Protocols for Scalable Multicore Platforms", in "Coordination Models and Languages (Proceedings of COORDINATION 2014)", LNCS 8459, pp. 65--82, 2014

[3] S.-S. Jongmans, F. Arbab: "Toward Sequentializing Overparallelized Protocol Code", in "proc. of ICE 2014", EPTCS, 2014

Please contact:

Farhad Arbab, Frank de Boer,
Sung-Shik Jongmans
CWI, The Netherlands
Tel: +31 20 592 [4056, 4139, 4241]
E-mail: [farhad, frb, jongmans]@cwi.nl

European Research and Innovation

InterpreterGlove - An Assistive Tool that can Speak for the Deaf and Deaf-Mute

by Péter Mátételki and László Kovács

InterpreterGlove is an assistive tool for hearing- and speech-impaired people that enables them to easily communicate with the non-disabled community through the use of sign language and hand gestures. Put on the glove, put the mobile phone in your pocket, use sign language and let speak for you!

Many hearing- and speech-impaired people use sign, instead of spoken, languages to communicate. Commonly, the only sectors of the community fluent in these languages are the affected individuals, their immediate friend and family groups and selected professionals.

We have created an assistive tool, InterpreterGlove, that can reduce communication barriers for the hearing- and speech-impaired community. The tool consists of a hardware-software ecosystem that features a wearable motion-capturing glove and a software solution for hand gesture recognition and text- and language-processing. Using these elements it can operate as a simultaneous interpreter, reading the signed text of the glove wearer aloud (Figure 1).

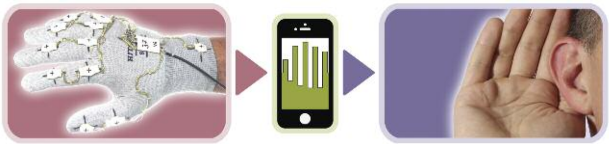


Figure 1: The key components of InterpreterGlove are motion-capturing gloves and a mobile-supported software solution that recognises hand gesture and completes text- and language-processing tasks.

Prior to using the glove it needs to be configured and adapted to the user's hand. A series of hand poses are recorded including bended, extended, crossed, closed and spread fingers, wrist positions and absolute hand orientation. This allows the glove to generate the correct gesture descriptor for any hand state. This personalization process not only enables the glove to translate every hand position into a digital handprint but also ensures that similar hand gestures will result in similar gesture descriptors across all users. InterpreterGlove is then ready for use. A built-in gesture alphabet, based on the international Dactyl sign language (also referred to as fingerspelling), is provided. This alphabet includes 26 one-handed signs, each representing a letter of the English alphabet. Users can further customize this feature by fine-tuning the pre-defined gestures. Thus, the glove is able to recognize and read aloud any fingerspelled word.

Feedback suggested that fingerspelling long sentences during a conversation could become cumbersome. However,

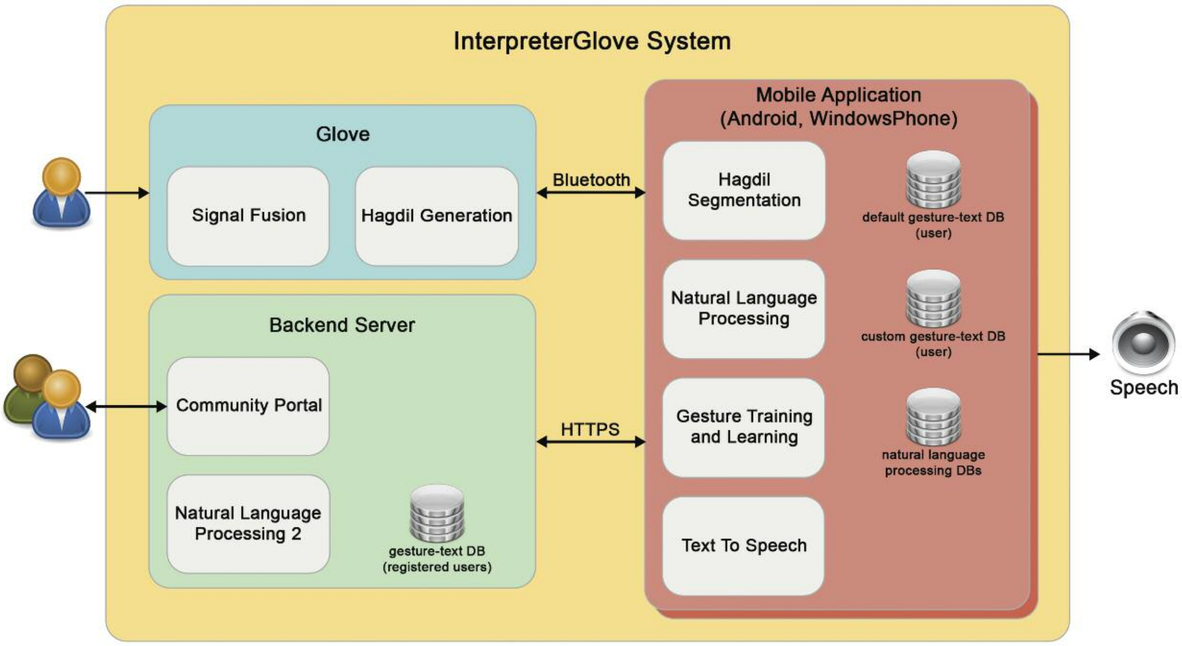

InterpreterGlove System
Glove
Signal Fusion
Hagdil Generation
Bluetooth
Mobile Application
(Android, WindowsPhone)
Hagdil Segmentation
default gesture-text DB (user)
Natural Language Processing
custom gesture-text DB (user)
Gesture Training and Learning
natural language processing DBs
Text To Speech
Backend Server
Community Portal
Natural Language Processing 2
gesture-text DB (registered users)
HTTPS
Speech

Figure 2: The building blocks of the InterpreterGlove ecosystem.

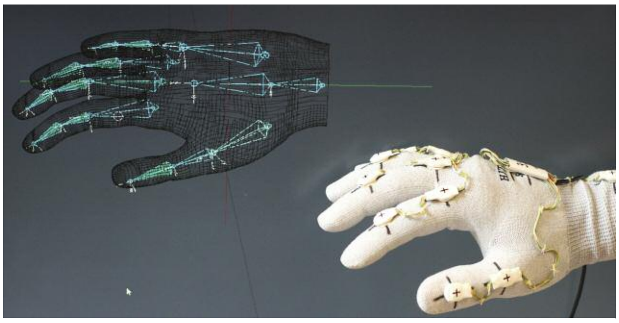
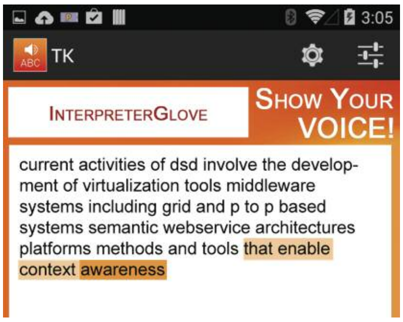


Figure 3: Hand gestures are transmitted to the mobile application using Hagdil gesture descriptors.



current activities of dsd involve the development of virtualization tools middleware systems including grid and p to p based systems semantic webservice architectures platforms methods and tools that enable context awareness

Figure 4: Two algorithms are applied to complete the segmentation and natural language processing components necessary for transforming hand gestures into understandable text. These are both intergrated into the mobile application.

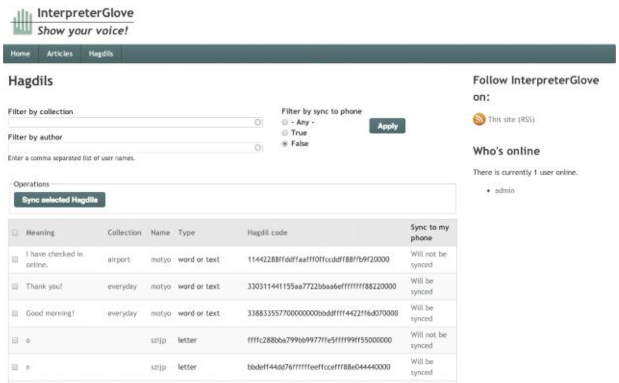
the customization capabilities allow the user to eliminate this inconvenience. Users can define their own gesture alphabet by adding new, customised gestures and they also have the option of assigning words, expressions and even full sentences to a single hand gesture. For example, “Good morning”, “Thank you” or “I am deaf, please speak slowly” can be delivered with a single gesture.

The main building blocks of the InterpreterGlove ecosystem are the glove, the mobile application and a backend server that supports the value-added services (Figure 2). The glove’s prototype is made of a breathable elastic material and the electronic parts and wiring are ergonomically designed to ensure seamless daily use. We used data from twelve ‘9 DoF (Degree of Freedom)’ integrated motion-tracking sensors to calculate the absolute position of the hand and to determine the joints’ deflections in 3D. The glove creates a digital copy of the hand which is denoted by our custom gesture descriptor. The glove connects to the user’s cell phone and transmits these gesture descriptors via a Bluetooth serial interface to the mobile application to be processed by the high-level signal- and natural-language processing algorithms.

Based on the biomechanical characteristics and kinematics of the human hand [1], we defined the semantics of Hagdil, the gesture descriptor. We use this descriptor to communicate the users’ gestures to the mobile device (Figure 3). Every second, 30 Hagdil descriptors are generated by the glove and transmitted to the mobile application.

Two types of algorithm are applied on the generated Hagdil descriptor stream to transform it into understandable text (Figure 4). To begin with, raw text is generated as a result of the segmentation, by finding the best gesture descriptor candidates. To achieve this a signal processing algorithm is required. Based on our evaluation, the sliding window and dynamic kinematics based solutions produced the best results and consequently these have been used in our prototype. This raw text may contain spelling errors caused by the user’s inaccuracy and natural signing variations. To address this issue, a second algorithm which performs an auto-correction function processes this raw text and transforms it into understandable words and sentences. This algorithm is based on a customised natural language processing solution that incorporates 1- and 2-gram database searches and a confusion matrix based weighting. The corrected output can then be read aloud by the speech synthesizer. Both algorithms are integrated into the mobile-based software application.

Our backend server operates as the central node for value-added services. It offers a community portal that facilitates gesture sharing between users (Figure 5). It also supports higher-



Hagdils

Filter by collection

Filter by author

Enter a comma separated list of user names.

Filter by sync to phone
○ - Any -
○ True
● False

Operations

Follow InterpreterGlove on:

This site (RSS)

Who's online

There is currently 1 user online.

- admin

□	Meaning	Collection	Name	Type	Hagdil code	Sync to my phone
□	I have checked in online.	airport	motyo	word or text	114422B8ffddffaafff0ffccddff88ffb9f20000	Will not be synced
□	Thank you!	everyday	motyo	word or text	330311441155aa7722bbaa6effffffff88220000	Will be synced
□	Good morning!	everyday	motyo	word or text	338833557700000000bbddffff4422ff6d070000	Will be synced
□	o		szijp	letter	ffffc288bba799bb9977ffe5ffff99ff55000000	Will not be synced
□	n		szijp	letter	bbdeff44dd76ffffffeeffcceffff88e044440000	Will be synced

Figure 5: The InterpreterGlove portal allows users to share their customised gestures.

level language processing capabilities than available from the offline text processing built into the mobile application.

Throughou t the whole project, we worked closely with the deaf and blind community to ensure we accurately captured their needs and used their expertise to ensure the prototypes of InterpreterGlove were properly evaluated. Their feedback deeply influenced our work and achievements. We hope that this device will improve and expand communication opportunities for hearing- and speech-impaired people and thus, enhance their social integration into the wider community. It may also boost their employment possibilities.

Although originally targeted to meet the needs of hearing-impaired people, we have also realised that this tool has considerable potential for many others, for example, those with a speech-impairment, physical disability or being rehabilitated following a stroke. We plan to address additional, targeted needs in future projects. Presently, two major areas have been identified for improvement that may have huge impacts on the application-level possibilities of this complex system. Integrating the capability to detect dynamics, i.e., perceive the direction and speed of finger and hand movements, opens up new interaction possibilities. Expanding the coverage of motions capture, by including additional body parts, opens the door for more complex application scenarios.

The InterpreterGlove ("Jelnyelvi tolmácskesztyű fejlesztése" KMR_12-1-2012-0024) project was a collaborative effort between MTA SZTAKI and Euronet Magyarország Informatikai Zrt. that ran between 2012 and 2014. The project was supported by the Hungarian Government, managed by the National Development Agency and financed by the Research and Technology Innovation Fund.

Links:
http://dsd.sztaki.hu/projects/TolmacsKesztyu
Dactyl fingerspelling:
http://www.lifeprint.com/ASL101/fingerspelling/index.htm

Reference:

[1] C. L. Taylor, R. J. Schwarz: "The Anatomy and Mechanics of the Human Hand", Artificial limbs 06/1955; 2(2):22-35.

Please contact:

Péter Mátételki or László Kovács, SZTAKI, Hungary
E-mail: {peter.matetelki, laszlo.kovacs}@sztaki.mta.hu

The OFSE-Grid: A Highly Available and Fault Tolerant Communication Infrastructure based on Openflow

by Thomas Pfeiffenberger, Jia Lei Du and Pedro Bittencourt Arruda

The project OpenFlow Secure Grid (OFSE-Grid) evaluates the use of a software-defined networking (SDN) infrastructure in the domain of energy communication networks.

Worldwide, electrical grids are developing into smart grids. To ensure reliability, robustness and optimized resource usage, these grids will need to rely heavily on modern information and communication technologies. To support the achievement of these goals in communication networks, we evaluated the possibility of using a software-defined networking (SDN) infrastructure based on OpenFlow, to provide a dependable communication layer for critical infrastructures.

SDN proposes a physical separation of the control and data planes in a computer network (Figure 1). In this scenario, only the controller is able to configure forwarding rules in the data plane of the switches. This has the advantage of giving the system a comprehensive and complete overview of itself. With this multifaceted knowledge about the status of the network, it is easier to implement new applications in the network by writing an application that configures it properly.

The implementation of a robust and fault-tolerant, multi-cast forwarding scheme based on an OpenFlow network architecture is one of the main goals of the OFSE-Grid project. To solve this issue we use a two-layered approach. To begin with, one must know how to forward packets correctly in the topology and then, decide what to do when a fault occurs. Different approaches have been published regarding how best to calculate the multicast tree for a network and make further improvements [1]. Fault-tolerance can be achieved either reactively or proactively. In a reactive fault-tolerant scheme, the SDN controller is responsible for recalculating the configuration rules when a failure happens. In a proactive fault-tolerant scheme, the controller pre-emptively installs all rules necessary for managing a fault.

In terms of robustness and the rational use of switch resources, a hybrid approach to fault-tolerance is best. Therefore, we propose making the network proactively tolerant to one fault (as in our current solution) so that there is very little packet loss on disconnection. However, we also propose that further research should be undertaken so that a network that is capable of reconfiguring itself to the new topology after the failure can be developed. Using this technique, the network is not only tolerant to a fault, but it is also

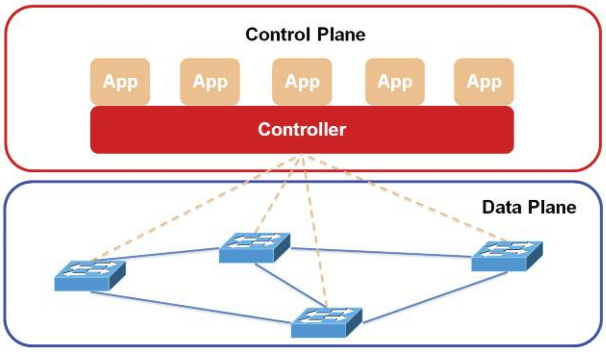


Figure 1: A software defined networking architecture.

able to maintain fault-tolerance after a fault. This is similar to the approach in [2] but here, we take advantage of local fault recovery which reduces the failover times and thus, packet loss during failovers. Of course, the algorithm controlling the network must run fast enough to avoid that a second failure happening before the network is reconfigured. If a situation in which two failures can occur almost simultaneously is expected, it would be advisable to make the network two-fault-tolerant. This can be achieved with minor modifications of our software but comes at a greater cost in terms of hardware resources, both in the controller and the involved network devices.

To verify our approach, we chose a topology that could approximate a critical network infrastructure such as a substation (Figure 2). The topology consists of multiple rings connected to a backbone ring. It is a fault-tolerant, multi-cast scenario and the configured forwarding rules are shown. The reconfigured multi-cast scenario after a link failure is shown in Figure 3. This new multi-cast tree is not simply a workaround to get to t1, but actually a whole new multi-cast tree.

As part of the OFSE-Grid project we also confirmed that in general, it will be possible to use commercial off-the-shelf SDN/OpenFlow hardware to provide a robust communication network for critical infrastructures in the future [3]. Looking forward, one of our next steps will be to consider latency and bandwidth requirements in the routing decisions as this may be a major precondition for critical infrastructure.

This work was part of the Open Flow Secure Grid (OFSE-Grid) project funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) within the "IKT der Zukunft" program.

Link:
http://www.salzburgresearch.at/en/projekt/ofse_grid_en/

References:

[1] H. Takahashi, A. Matsuyama: "An approximate solution for the Steiner problem in graphs", Math. Japonica, vol. 24, no. 6, 1980

[2] D. Kotani, K. Suzuki, H. Shimonishi: "A Design and Implementation of OpenFlow Controller Handling IP Multicast with Fast Tree Switching," IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT), 2012

[3] T. Pfeiffenberger, J. L. Du: "Evaluation of Software-Defined Networking for Power Systems", IEEE International Conference on Intelligent Energy and Power Systems (IEPS), 2014

Please contact:

Thomas Pfeiffenberger
Salzburg Research Forschungsgesellschaft mbH
E-mail: thomas.pfeiffenberger@salzburgresearch.at

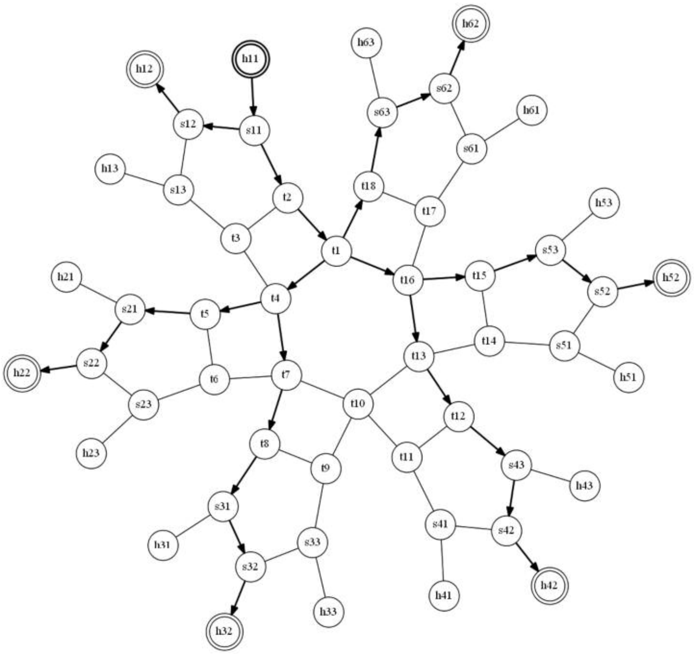


Figure 2: A 2-approximation calculation of the optimal Steiner tree for the multi-cast group of the topology.

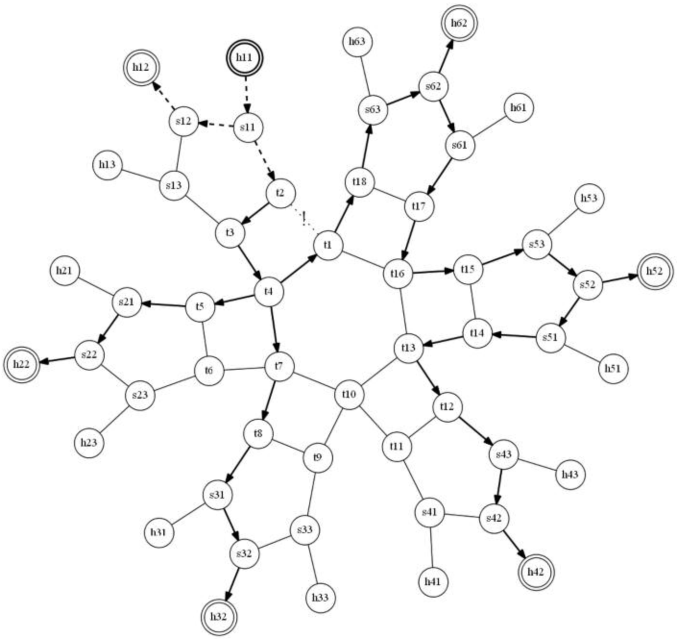


Figure 3: Network behaviour when the link t2 - t1 fails. When this happens, the switch forwards the packet to a different tree (bold edges), which can be used to forward packets to the destinations without using the faulty link.

Learning from Neuroscience to Improve Internet Security

by Claude Castelluccia, Markus Duermuth and Fatma Imamoglu

This project, which is a collaboration between Inria, Ruhr-University Bochum, and UC Berkeley, operates at the boundaries of Neuroscience and Internet Security with the goal of improving the security and usability of user authentication on the Internet.

Most existing security systems are not user friendly and impose a strong cognitive burden on users. Such systems usually require users to adapt to machines, whereas we think that machines should be adjusted to users. There is often a trade-off between security and usability: in current applications security tends to decrease usability. A prime example for this trade-off can be observed in user authentication, which is an essential requirement for many web sites that need to secure access to stored data. Most Internet services use password-authentication based schemes for user authentication.

Password-authentication based schemes are knowledge-based, since they require users to memorize secrets, such as passwords. In password-based authentication schemes, higher security means using long, random combination of characters as passwords, which are usually very difficult to remember. In addition, users are asked to provide different passwords for different web-sites, which have their own specific policy. These trade-offs are not well understood, and password-based authentication is often unpopular among users [1]. Despite substantial research focusing on improving the state-of-the-art, very few alternatives are in use.

This project explores a new type of knowledge-based authentication scheme that eases the high cognitive load of passwords. Password-based schemes, as well as other existing knowledge-based authentication schemes, use explicit memory. We propose a new scheme, MooneyAuth, which is based on implicit memory. In our scheme, users can reproduce an authentication secret by answering a series of questions or performing a task that affects their subconscious memory. This has the potential to offer usable, deployable, and secure user authentication. Implicit memory is effortlessly utilized for every-day activities like riding a bicycle or driving a car. These tasks do not require explicit recall of previously memorized information.

The authentication scheme we propose is a graphical authentication scheme, which requires users to recognize Mooney images, degraded two-tone images that contain a hidden object [2]. In contrast to existing schemes, this scheme is based on visual implicit memory. The hidden object is usually hard to recognize at first sight but is easy to recognize if the original image is presented beforehand (see Figure 1). This technique is named after Craig Mooney, who used similar images of face drawings as early as 1957 to study the perception of incomplete pictures in children [3].

Our authentication scheme is composed of two phases: In the priming phase, the user is ‘primed’ with a set of images, their Mooney versions and corresponding labels. During the authentication phase, a larger set of Mooney images, including the primed images from the priming phase, is displayed to the user. The user is then asked to label the Mooney images that she was able to recognize. Finally, the system computes an authentication score from the correct and incorrect labels and decides to grant or deny access accordingly. A prototype of our proposed authentication scheme can be found online (see link below). We tested the viability of the scheme in a user study with 230 participants. Based on the participants from the authentication phase we measured the performance of our scheme. Results show that our scheme is close to being practical for applications where timing is not overly critical (e.g., fallback authentication).

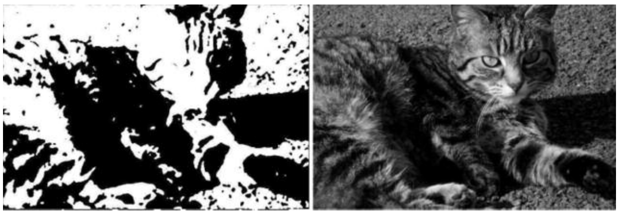


Figure 1: Left is the modified gray-scale version of the image, right is the Mooney version of the gray-scale image [2].
Copyright for the original image by Alex Pepperhill (CC by 2.0, source: https://www.flickr.com/photos/56278705@N05/8854256691/in/photostream/).

We believe that this line of research, at the frontier of cognitive neuroscience and Internet security, is very promising and requires further research. In order to improve the usability of authentication schemes, security researchers must achieve a better understanding of human cognition.

Link:
http://www.mooneyauth.org

References:

[1] A. Adams and M. A. Sasse: “Users are not the enemy”, Commun. ACM, 42(12):40-46, Dec. 1999
[2] F. Imamoglu, T. Kahnt, C. Koch and J.-D. Haynes: “Changes in functional connectivity support conscious object recognition”, NeuroImage, 63(4):1909-1917, Dec. 2012
[3] C. M. Mooney: “Age in the development of closure ability in children”. Canadian Journal of Psychology, 11(4):219-226, Dec. 1957.

Please contact:

Claude Castelluccia
Inria, France
E-mail: claude.castelluccia@inria.fr

Markus Duermuth
Ruhr-University Bochum, Germany
E-mail: markus.duermuth@rub.de

Fatma Imamoglu
UC Berkeley, U.S.A.
E-mail: fatma@berkeley.edu

Mathematics Saves Lives: The Proactive Planning of Ambulance Services

by Rob van der Mei and Thije van Barneveld

Research into the fields of information and communications technology and applied mathematics is very relevant for today's society. At CWI, we have pursued investigations in these fields to enhance the vital services provided by ambulances through the development of Dynamic Ambulance Management.

In life-threatening emergency situations, the ability of ambulance service providers (ASPs) to arrive at an emergency scene within a short period of time can make the difference between survival or death for the patient(s). In line with this, a service-level target is commonly used that states that for high-emergency calls, the response time, i.e., the time between an emergency call being placed and an ambulance arriving at the scene, should be less than 15 minutes in 95% of cases. To realise such short response times, but still ensure running costs remain affordable, it is critical to efficiently plan ambulance services. This encompasses a variety of planning problems at the tactical, strategic and operational levels. Typical questions that must be answered include "How can we reliably predict emergency call volumes over time and space?", "How can we properly anticipate and respond to peaks in call volumes?", "How many ambulances are needed and where should they be stationed?", and "How should ambulance vehicles and personnel be effectively scheduled?".

A factor that further complicates this type of planning problems is uncertainty, an ever-present consideration that in this context, will affect the entire ambulance service-provisioning process (e.g., emergency call-arrival patterns, travel times, etc.). The issue is that the planning methods currently available typically assume that "demand" (in the context of ambulance services this would refer to call volumes and their geographical spread) and "supply" (the availability of vehicles and ambulance personnel at the right time in the right place) parameters are known a priori. This make these methods highly vulnerable to randomness or uncertainty, and the impacts this inevitably has on the broader planning process, namely inefficiencies, and higher costs. For ambulance services, the challenge is to develop new planning methods that are both scalable and robust against the inherent randomness associated with the service process, both real and non-real time.

A highly promising development that is gaining momentum in the ambulance sector is the emergence of Dynamic Ambulance Management (DAM). The basic idea of DAM is that ambulance vehicles are proactively relocated to achieve a good spatial coverage of services in real time. By using dynamic and proactive relocation strategies, shorter arrival times can be achieved [1].

To illustrate the use of DAM, consider the following example area which features six cities (A, D, E, L, U and Z; Figure 1), serviced by seven ambulances. When there are no emergencies (a 'standard situation'), optimal vehicle coverage is obtained by positioning one ambulance in each of the six cities, with one additional vehicle in city A as it has the largest population. Now consider a scenario where an incident occurs at city L while simultaneously, two additional incidents are occurring in city A. These incidents can all be serviced by the ambulances currently located in those two cities. Under an 'optimal' dynamic relocation policy, this scenario should then trigger a proactive move by the ambulance in city D to city L, in order to maintain service coverage. As soon as that ambulance is within a 15-minute driving range of city L, the ambulance at city U should move to city D (note that city U is smaller than D, meaning that it can be covered by city E's ambulance). The ambulance at city Z can then proactively move to city A. This example illustrates the complexity in using DAM: for example, what additional steps would be appropriate if an additional accident was to occur in city U whilst the ambulance was transitioning between cities U and D?

The key challenge in an approach such as DAM is developing efficient algorithms that support real-time decision making. The fundamental question is "under what circumstances should proactive relocations be performed, and how effective are these relocation actions?". Using methods from the stochastic optimization techniques Markov Decision Processes and Approximate Dynamic Programming, we have developed new heuristics for DAM [2,3]. Implementation of these solutions in the visualization and simulation package suggests that strong improvements in service quality can be realised, when compared with the out-

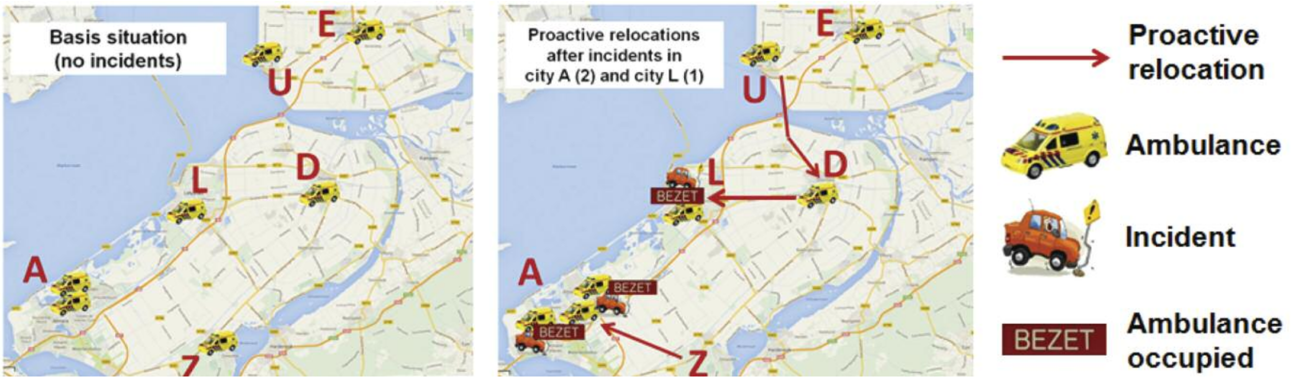

Basis situation (no incidents)
E
U
L
D
A
Z
Proactive relocations after incidents in city A (2) and city L (1)
E
U
L
D
A
Z
BEZET
BEZET
BEZET
Proactive relocation
Ambulance
Incident
BEZET
Ambulance occupied

Figure 1: Illustration of proactive relocations of ambulance vehicles.

comes gained from the relocation rules currently being deployed by most ambulance providers. Currently we are in the field trial phase and are setting up a number of newly developed DAM algorithms, in collaboration with a number of ASPs in the Netherlands. These activities are part of the project 'From Reactive to Proactive Planning of Ambulance Services', partly funded by the Dutch agency Stichting Technologie & Wetenschap.

Links:
http://repro.project.cwi.nl
Dutch movie: http://www.wetenschap24.nl/programmas/de-kennis-van-nu-tv/onderwerpen/2014/april/wiskunde-redt-mensenlevens.html

References:
[1] M. S. Maxwell et al.: "Approximate dynamic programming for ambulance redeployment", INFORMS Journal on Computing 22 (2) (2010) 266-281
[2] C.J. Jagtenberg, S. Bhulai, R.D. van der Mei: "A polynomial-time method for real-time ambulance redeployment", submitted.
[3] T.C. van Barneveld, S. Bhulai, R.D. van der Mei: "A heuristic method for minimizing average response times in dynamic ambulance management", submitted.

Please contact:
Rob van der Mei
CWI, The Netherlands
E-mail: mei@cwi.nl

An IoT-based Information System Framework towards Organization Agnostic Logistics: The Library Case

by John Gialelis and Dimitrios Karadimas

SELIDA, a printed materials management system that uses radio frequency identification (RFID), complies with the Web-of-Things concept. It does this by employing object naming based services that are able to provide targeted information regarding RFID-enabled physical objects that are handled in an organization agnostic collaborative environment.

Radio Frequency Identification (RFID) technology has already revolutionised areas such as logistics (i.e., supply chains), e-health management and the identification and traceability of materials. The challenging concept of RFID-enabled logistics management and information systems is that they use components of the Electronic Product Code (EPC) global network, such as Object Naming Services (ONS) and the EPC Information Services (EPCIS) in order to support the Internet of Things concept.

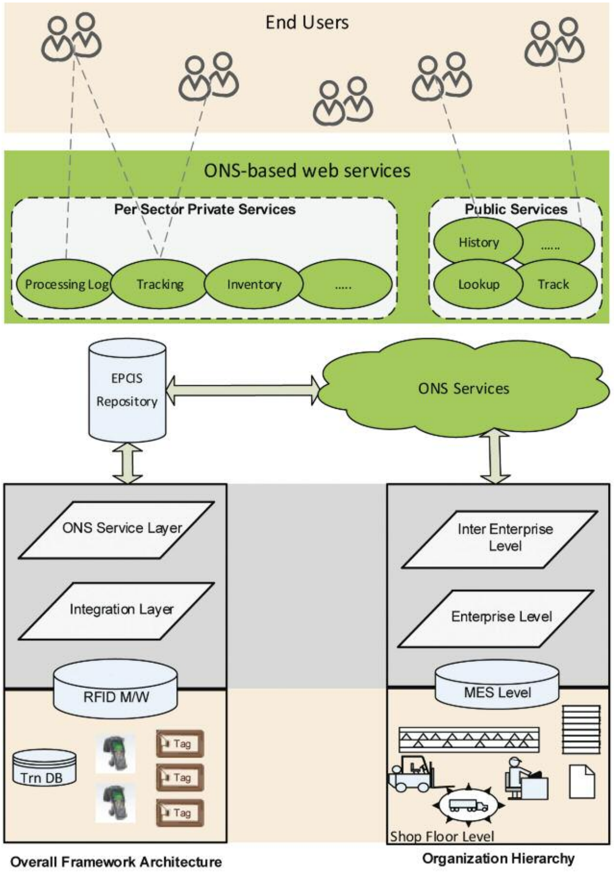


Figure 1: The architectural framework of SELIDA.

SELIDA is a joint research project between the Industrial System Institute, the University of Patras (Library and Information Center), the Athens University of Economics and Business, Ergologic S.A and Orasys ID S.A. This project introduces an architectural framework that aims to support as many of the EPC global standards as possible (Figure 1). The project's main goal is the ability to map single physical objects to URIs in order to provide, to all involved organizations in the value chain, various information related to these objects (tracking, status, etc). This is mainly achieved by SELIDA's architectural framework which is able to support as many of the EPC global standards as possible (Figure 1) along with the realization of ONS-based web services available in the cloud. This architectural framework is a value-chain agnostic which relates to:
- the common logistics value-chain;
- the physical documents inter-change value-chain; and
- in demanding cases, the objects inter-change value-chain.

The discovery and tracking service of physical documents that has been implemented exploits both ONS 1.0.1 and EPCIS 1.0.1, in order to allow EPC tagged documents to be mapped to the addresses of arbitrary object management services (OMS), albeit ones with a standardised interface.

The main constituents of the architectural framework are:
- The RFID middleware which is responsible for receiving, analysing processing and propagating the data collected by the RFID readers to the information system which supports the business processes.
- The Integration Layer which seamlessly integrates the EPC related functions to the existing services workflow.

While the existing legacy systems could be altered, such a layer is preferable because of the reliability offered by shop floor legacy systems in general.

- The ONS Resolver which provides secure access to the ONS infrastructure so that its clients can not only query the OMSs related to EPCs (which is the de facto use case for the ONS) but also introduce new OMSs or delete any existing OMSs for the objects.
- The OMS which provides management, tracking and other value added services for the EPC tagged objects. The ONS Resolver maps the OMS to the objects, according to their owner and type, and they should be implemented according to the EPCIS specification (see link below).

The SELIDA architecture has been integrated into KOHA, the existing Integrated Library System used in the University of Patras Library and Information Center. As with all integrated library systems, KOHA supports a variety of workflows and services that accommodate the needs of the Center. The SELIDA scheme focuses on a handful of those services and augments them with additional features. This is generally done by adding, in a transparent way, the additional user interface elements and background processes that are needed for the scheme to work. In order to provide the added EPC functionality to the existing KOHA operations, the integration layer was designed and implemented to seamlessly handle all the extra work, along with the existing service workflow. The SELIDA scheme provides additional functionality to services such as Check Out, Check In, New Record and Delete Record. There are also a number of tracking services that our scheme aims to enhance; these are History, Location and Search/Identify.

The implemented architecture focuses on addressing the issue of empowering the whole framework with a standard specification for object tracking services by utilising an ONS. Thus, the organisations involved are able to act agnostically of their entities, providing them with the ability to resolve EPC tagged objects to arbitrary services in a standardised manner.

Links:
KOHA: www.koha.org
ISO RFID Standards: http://rfid.net/basics/186-iso-rfid-standards-a-complete-list
Survey: http://www.rfidjournal.com/articles/view?9168
EPCglobal Object Name Service (ONS) 1.0.1:
http://www.gs1.org/gsmp/kc/epcglobal/ons/ons_1_0_1-standard-20080529.pdf
EPCglobal framework standards:
http://www.gs1.org/gsmp/kc/epcglobal

Reference:

[1] J. Gialelis, et al.: An ONS-based Architecture for Resolving RFID-enabled Objects in Collaborative Environments", IEEE World Congress on Multimedia and Computer Science, WCMCS 2013

Please contact:

John Gialelis or Dimitrios Karadimas
Industrial Systems Institute, Patras, Greece
E-mail address: {gialelis,karadimas}@isi.gr

Lost Container Detection System

by Massimo Cossentino, Marco Bordin and Patrizia Ribino

Each year thousands of shipping containers fail to arrive at their destinations and the estimated damage arising from this issue is considerable. In the past, a database of lost containers was established but the difficult problem of identifying them in huge parking areas was entrusted to so-called container hunters. We propose a system (and related methods) that aims to automatically retrieve lost containers inside a logistic area using a set of sensors that are placed on cranes working within that area.

Figure 1: A common shipping area (by courtesy of Business Korea).

The Lost Container Detection System (LostCoDeS) [1] is an ICT solution created for avoiding the costly loss of containers inside large storage areas, such as logistic districts or shipping areas. In these kinds of storage areas (Figure 1), several thousand of containers are moved and stacked in dedicated zones (named cells) daily. Nowadays, the position of each stacked container is stored in a specific database that facilitates the later retrieval of this location information. As the movement and management of containers involves many different workers (e.g., crane operators, dockers, administrative personnel, etc.), communication difficulties or simply human distraction can cause the erroneous positioning of containers and/or the incorrect updating of location databases. In large areas that store thousands of containers, such errors often result in containers becoming lost and thus, result in the ensuing difficulties associated with finding them.

At present, to the best of our knowledge, there are no automatic solutions available that are capable of solving this particular problem without the pervasive use of tracking devices. Most of the proposed solutions in the literature address container traceability during transport (either to their destinations or inside logistic districts) by using on-board tracking devices [2] or continuously monitoring the containers with ubiquitous sensors [3], only to name a few.

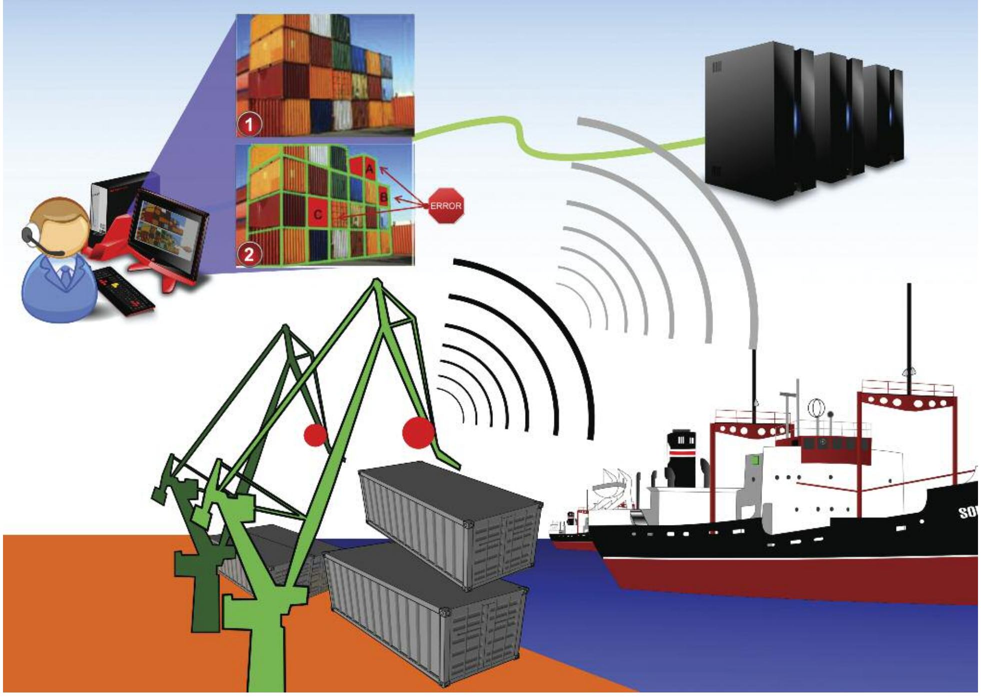


Figure 2: An informal architectural representation of the Lost Container Detection System (LostCoDeS).

The LostCoDeS is a system that is able to detect a misplaced container inside a large storage area without using any kind of positioning or tracking devices on the container. The novelty of the LostCoDeS lies in the method we use to find the lost containers, rather than on the wide use of hardware devices on containers. In this system, a few sets of sensors are placed on the cranes working inside the logistic area. Using the data from these sensor sets, an algorithm can then verify if there are any misplaced containers that may indicate a lost item. The architectural design of LostCoDeS is quite simple. It is composed of a set of sensors for capturing geo-data related to the large storage area, a workstation for elaborating these data and a network for communicating data to a storage device. An informal architectural representation of LostCoDeS is presented in Figure 2.

From the functional perspective, LostCoDeS is based on three main algorithms: the former allows to execute a multi-modal fusion of data coming from the set of sensors; the second one is able to reproduce a three-dimensional representation of geo-data and finally the last one implements a comparison between real data perceived by the sensors and expected ones.

Hence, our system is able to generate a representation of the container stacks to detect anomalies.More in detail, the LostCoDeS is able to (i) detect the incorrect placement of containers; (ii) identify the likely locations of lost containers; (iii) indicate the presence of non-registered containers; (iv) indicate the absence of registered containers; and (v) monitor the positioning operations.

The main advantage of the LostCoDeS is that the detection of lost containers can be completed during normal handling operations. Moreover, it overcomes the limitations associated with traditional tracking systems which are based on radio signals, namely that the reliability and stability of transmissions is not guaranteed when the signal has to pass through obstacles (e.g., metal). Further, the system is discrete (i.e., there is no need to install cameras and/or other equipment over the monitored area) and low cost (i.e., there is no need to install tracking devices on the containers), but still maintains an ability to monitor large areas. It is also worth noting that the continuous monitoring of operations is not strictly required (although useful) as the system is capable of identifying misplaced containers even when it starts from an unknown location.

Acknowledgements

A special thank to Ignazio Infantino, Carmelo Lodato, Salvatore Lopes and Riccardo Rizzo who along with the authors are inventors of LostCoDeS.

References:

[1] M. Cossentino et al: “Sistema per verificare il numero di contenitori presenti in una pila di contenitori e metodo di verifica relativo”, Italian Pending Patent n. RM2013A000628, 14 Nov 2013

[2] Cynthia Marie Braun: “Shipping container monitoring and tracking system.” U.S. Patent No. 7,339,469. 4 Mar. 2008

[3] Leroy E. Scheppmann: “Systems and methods for monitoring and tracking movement and location of shipping containers and vehicles using a vision based system.” U.S. Patent No. 7,508,956. 24 Mar. 2009.

Please contact:
Patrizia Ribino, ICAR-CNR, Italy
E-mail: ribino@pa.icar.cnr.it

Smart Buildings: An Energy Saving and Control System in the CNR Research Area, Pisa

by Paolo Barsocchi, Antonino Crivello, Erina Ferro, Luigi Fortunati, Fabio Mavilia and Giancarlo Riolo

"Renewable Energy and ICT for Sustainability Energy" (or "Energy Sustainability") is a project led by the Department of Engineering, ICT, and Technologies for Energy and Transportations (DIITET) of the Italian National Research Council (CNR). This project aims to study and test a coordinated set of innovative solutions to make cities sustainable, with respect to their energy consumption.

To achieve its goal, the project is based on i) the widespread use of renewable energy sources (and related storage technologies and energy management), ii) the extensive use of ICT technologies for the advanced management of energy flows, iii) the adaption of energy-efficient city services to demands (thus encouraging rationale usage of energy resources and, thus, savings), and iv) the availability of energy from renewable sources.

This project is aligned with the activities of the European Commission under their energy efficiency theme. By June 2014, the European Member States will have to implement the new Directive 2012/27/EU (4 December 2012). This Directive establishes that one of the measures to be adopted is "major energy savings for consumers", where easy and free-of-charge access to data on real-time and past energy consumption, through more accurate individual metering, will empower consumers to better manage their energy consumption [1].

The Energy Sustainability project focuses on six activities. We are involved in the "in-building" energy sustainability component. The main goal of this sub-project is to compute the real energy consumption of a building, and to facilitate energy savings when and where possible, through the experimental use of the CNR research area in Pisa. This research area is more complex than a typical, simple building, as it hosts 13 institutes. However, energy to the area is supplied through a single energy source, which means that the institutes have no way of assessing their individual energy consumption levels.

The main requirements of the In-Building sub-project are that it must:

- monitor the power consumption of each office (lights and electrical sockets),
- regulate the gathering and visualization of data via permits,
- support real-time monitoring and the ability to visualize time series,
- define energy saving policies,
- be cheap and efficient.

We developed an Energy long-term Monitoring System (hereafter referred to as the EMS@CNR), which is composed of a distributed ZigBee Wireless Sensor Network (WSN), a middleware communication platform [2] and a set of decision policies distributed on the cloud (Figure 1). At the time of writing, there are sensor nodes of this WSN installed in some offices of our Institute (CNR-ISTI). Each sensor node can aggregate multiple transducers such as humidity, temperature, current, voltage, Passive Infrared (PIR), pressure sensors and noise detector. Each node is connected to a ZigBee Sink, which provides Internet-of-Things connectivity through the IPv6 addressing methodology. The choice to use a ZigBee network was driven by several technology characteristics, such as ultra low-power consumption, the use of unlicensed radio bands, cheap and easy installation, flexible and extendable networks, integrated intelligence for network set-up and message routing.

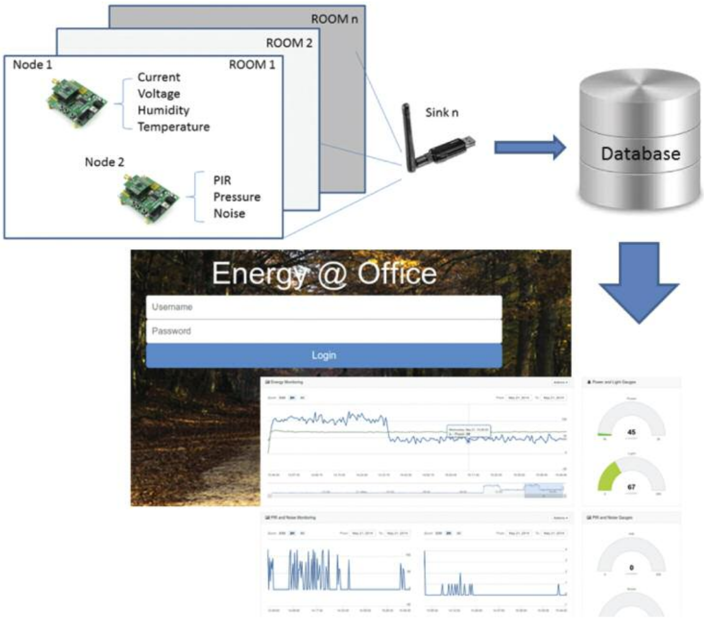


Figure 1: The EMS@CNR system and the WebOffice platform.

In order to measure the energy consumption of a room, we need to assess the values of current and voltage waveforms at the same instant. This is driven by the need to operate within existing buildings, without the possibility of changing existing electrical appliances. We used a current and voltage transformer. We also installed a PIR and a noise detector in a single node, and a set of pressure detectors, installed in another node under the floating floor, in order to determine the walking direction of a person (i.e., to detect if he is entering or leaving the office). This helps to determine whether or not someone is in the office, which is an information that drives decisions regarding potential energy savings for that specific room. As an example, in an office where nobody is present, lights and other electric appliances (apart from computers) can be automatically switched off. Currently, the decision policies do not take into account data coming from welfare transducers, such as temperature and relative humidity. This data will be included in the in-progress decision policies.

Sensor data collected by the deployed WSN are stored in a NoSQL document-oriented local database, such as

MongoDb. In order to provide both a real-time view of the sensor data and a configuration interface, we implemented a web interface, named WebOffice, that runs on JBoss 7.1. It implements the JavaEE specification and it is free and open-source. The Web interface provides a login system to protect data display and ensure privacy. After making a successful login, according with the permission, the sensor data are shown in the main page. There are two main types of graphics: dynamic charts (for real-time visualization and historical data) and gauges (for an immediate display of the last value recorded).

Links:
ZigBee: http://www.zigbee.org/
JBoss: http://www.jboss.org/overview/

References:

[1] L. Pérez-Lombard, J. Ortiz, C. Pout: “A review on buildings energy consumption information,” Energy and buildings, vol. 40, no. 3, pp. 394–398, 2008
[2] F. Palumbo et al.: “AAL Middleware Infrastructure for Green Bed Activity Monitoring”, Journal of Sensors, vol. 2013, pp. 1-15.

Please contact:

Erina Ferro, ISTI-CNR, Italy
E-mail: erina.ferro@isti.cnr.it

T-TRANS: Benchmarking Open Innovation Platforms and Networks

by Isidoros A. Passas, Nicos Komninos and Maria Schina

What might sixty web-based platforms and networks have to tell us about open innovation? The FP7 project T-TRANS aims to define innovation mechanisms for Intelligent Transport Systems (ITS) that facilitate the transfer of related innovative products and services to the market.

The T-TRANS project addresses the difficulties associated with transferring new technologies, and seeks to capitalise on the significant opportunities to improve efficiency and reduce costs once those technologies are commercialised. One of the expected outcomes of the project will be the establishment of a pilot innovation network that is focused on ITS. Initially this network will feature three glocal (global to local) communities that are suitable for ITS commercialisation, referred to as CIMs, and will be implemented in Central Macedonia (Greece), Galicia (Spain) and Latvia. This will set the scene for a more expansive Europe-wide ITS e-innovation network.

With the view to informing the design of this new e-innovation network, T-TRANS partners undertook an analysis to benchmark open innovation platforms and networks [1]. From this work, the partners were able to gain a better understanding of what it takes for a network to serve the objectives of its members and sustain itself effectively. Benchmarking is widely defined as the act of comparatively assessing an organisation’s technologies, production processes and products against leading organisations in similar markets. The T-TRANS benchmarking exercise was based on three main pillars of comparison and assessment: (1) the platform, (2) the collaboration network and (3) the added value of the network and the platform. Each of these pillars has been described by a set of characteristics or attributes. A benchmarking template was developed to capture the data that was included and the indicators that were used to benchmark each characteristic.

Platforms are considered to be those web-based systems that can be programmed and therefore, customised by developers and users. They may also accommodate the goals for on-going collaborative and/or joint initiatives that the original developers could not have possibly contemplated or had time to accommodate. A forthcoming publication by Angelidou et al. [2] presents an analysis of the current trends in innovation strategies set by the companies. The kind of trends that appear are 1) the majority of firms introducing new-to-market innovation do perform in-house R&D; 2) companies turn to open innovation and collaborative networks, especially the formation of global networks, and external knowledge partnerships for the acquisition of knowledge, fresh ideas and market access; 3) users and consumers also play a growing role, increasing the interaction between demand and supply; 4) multinational firms have a leading role in the globalisation of innovation; 5) local knowledge and capabilities as well as proximity to research and education institutions continue to matter for innovation.

A key finding of this work was that the results “indicate that there is no return to the old linear model of innovation, and R&D translates directly and spontaneously to innovation. On the contrary the systemic and network perspective is consolidated and shapes all drivers of innovation creation, such as universities and tertiary education, patenting and technology transfer, knowledge infrastructure and flows, international cooperation, governance and stakeholders’ involvement in shaping policies for innovation. Traditional innovation networks comprising only a few nodes evolve to extremely large networks with hundreds of participants from all over the world. They include local and global partners, but with the spread of ICTs and virtual networks they are becoming glocal, combining local competences with global know-how and access to markets”. This perspective provided the necessary definition framework for collaborative innovation networks and has been used in the T-TRANS benchmarking exercise.

Of the sixty cases considered in the benchmarking analysis, 66% were characterised as platforms and 51% as networks. The objectives of both platforms and networks are very clearly stated and identified. Clear objectives are crucial since they state exactly what the platform and/or network is intended to either build or support. Having clearly defined objectives supports the enrolment of new users and members and on-going operations. Some of the key objectives related to the collaborative design and development of products, problem solving, brainstorming and the creation of communities for crowdsourcing ideas. Interestingly, one of the plat-

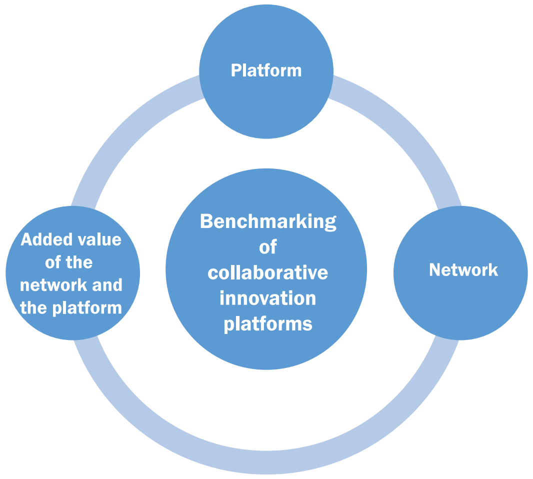

Platform
Added value of the network and the platform
Benchmarking of collaborative innovation platforms
Network

Figure 1: Open innovation platforms and networks benchmarking framework.

form/network combinations we examined provided a gamified community where members could complete missions and earn points and badges. Another identified objective was to support ideas that can hel p improve living conditions, including activities ranging from early stage investment to in-depth research thus, strengthening the social aspect of innovation and the development of innovative new products.

Following an analysis of these objectives, in accordance with the clear statements they made, we found that in the majority of instances the benefits of the platforms and networks we examined were clearly stated as well. Some of the benefits identified included effective cross-cultural collaboration, a deep understanding of complex issues, patent application and invention licencing coaching and collaboration opportunities in R&D and innovation. A commonly identified benefit was the free access to shared knowledge. In general, most of the networks stated that a creative process was much more powerful if it was fuelled by large numbers of participants who were all thinking about the same problem at the same time. The promotion of businesses between inventors and interested parties was another common benefit and crowdsourcing capabilities appeared to be an emerging benefit trend.

Most of the platforms were focused although not to a significant degree. Among the thematic domains identified were innovations services and patent invention support, grants and innovation management, disruptive and open innovation, innovation management and technology transfer and networking services.

The main supporting actions performed by the networks are towards knowledge transfer, collaboration and joint development. The platforms were categorised into 7 new product development stages as defined by the Coopers' Stage Gate methodology (Figure 2). The platforms mainly supported the processes which occur in the first four stages: idea generation, screening, concept development and testing and business analysis.

In conclusion, web-based platforms are an essential component of innovation networks, enlarging and extending collaborative opportunities across geographical and time zones and enabling the participation of large numbers of users, inventors and innovators, thus supporting an "innovation-for-all" culture [3].

Links:
Project T-TRANS: http://www.ttransnetwork.eu/
List of examined open innovation platforms and entities: http://wp.me/a2OwBG-PW
Stage-Gate innovation process: http://www.stage-gate.com/aboutus_founders.php

References:

[1] H. Chesbrough, W. Vanhaverbeke and J West J. (eds.): "Open Innovation: Researching a New Paradigm", Oxford University Press, 2006
[2] M. Angelidou, et al.: "Intelligent transport systems: Glocal communities of interest for technology commercialisation and innovation", 2014
[3] N. Komninos: "The Age of Intelligent Cities: Smart Environments and Innovation-for-all Strategies", London and New York, Routledge, 2014.

Please contact:

Isidoros Passas, INTELSPACE SA, Thessaloniki, Greece
E-mail: iapassas@intelspace.eu

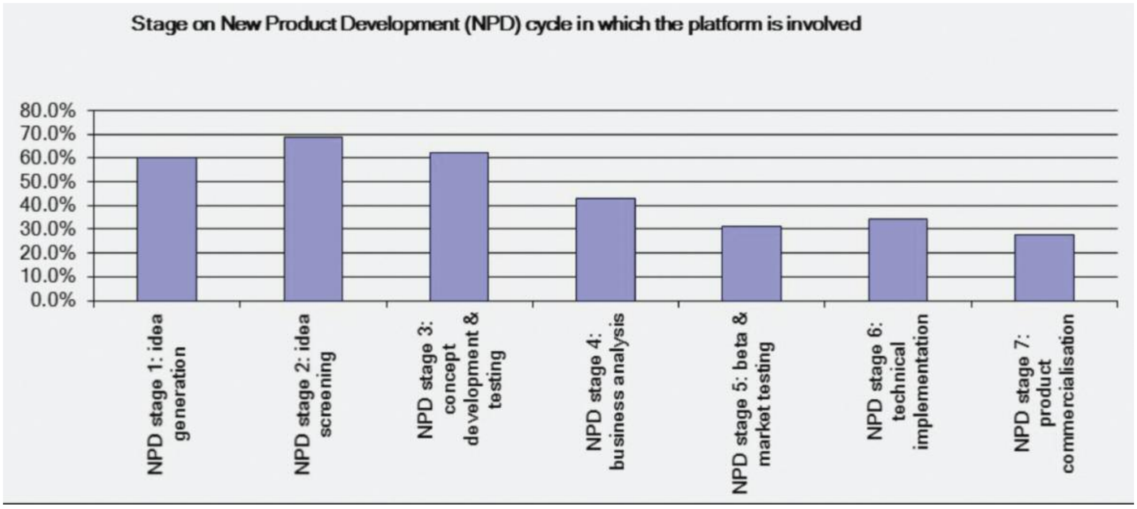


Figure 2: The platforms in the analysis were categorised according to which of the seven new product development stages they were involved.

CLOUDFLOW First Open Call for Application Experiments

CloudFlow - Computational Cloud Services and Workflows for Agile Engineering - is a European Integrating Project (IP) in the framework of Factories of the Future (FoF) that aims at making Cloud infrastructures a practical solution for manufacturing industries, preferably small and medium-sized enterprises (SMEs). The objective of CloudFlow is to ease the access to computationally demanding virtual product development and simulation tools, such as CAD, CAM, CAE, and make their use more affordable by providing them as Cloud services.

The project is now open for new (teams of) participants and solicits small consortia consisting of one to four partners (end users, software vendors, HPC/Cloud infrastructure providers and research organizations) to respond to the open call for proposals. With the call, the project seeks to increase the number of partners and application experiments currently being carried out within the CloudFlow project.

Application experiments will be rooted in computational technology for manufacturing and engineering industries, preferably SMEs, in stages covering but not limited to:

- design (CAD),
- simulation (product, process, factory, etc.),
- optimization,
- visualization,
- manufacturing planning,
- quality control and
- data management,

addressing workflows along the value chain in and across companies.

The deadline of the first Call is 30 September 2015. The expected duration of participation is January to December 2015. A second Call is expected to be launched in June 2015.

More information:
http://www.eu-cloudflow.eu
http://www.eu-cloudflow.eu/open-calls/first-call.html

7th International Conference of the ERCIM WG on Computational and Methodological Statistics

Pisa, Italy, 6-8 December 2014

The 7th International Conference on Computational and Methodological Statistics is organised by the ERCIM Working Group on Computational and Methodological Statistics and the University of Pisa.

The conference will take place jointly with the 8th International Conference on Computational and Financial Econometrics (CFE 2014). The conference has a high reputation of quality presentations. The last editions of the joint conference CFE-ERCIM gathered over 1200 participants.

Topics

Topics include all subjects within the aims and scope of the ERCIM Working Group CMStatistics: robust methods, statistical algorithms and software, high-dimensional data analysis, statistics for imprecise data, extreme value modeling, quantile regression and semiparametric methods, model validation, functional data analysis, Bayesian methods, optimization heuristics in estimation and modelling, computational econometrics, quantitative finance, statistical signal extraction and filtering, small area estimation, latent variable and structural equation models, mixture models, matrix computations in statistics, time series modeling and computation, optimal design algorithms and computational statistics for clinical research.

The journal Computational Statistics & Data Analysis will publish selected papers in special peer-reviewed, or regular issues.

More information:
http://www.cmstatistics.org/ERCIM2014/

Research Data Alliance and Global Data and Computing e-Infrastructure challenges

Rome, Italy, 11-12 December 2014,

The Research Data Alliance and global Data and Computing e-Infrastructure challenges event is being organised within the framework of the Italian Presidency of the European Union and will take place in Rome, Italy on 11-12 December 2014.

This event will focus on how synergies between e-Infrastructures and the ambitious European Research Infrastructures roadmap (ESFRI) and other major initiatives with high potential impact on research and innovation (e.g. HBP, COPERNICUS, and other initiatives across Horizon 2020) can be strengthened. This implies a strong European coordination between these initiatives. It also puts particular emphasis on the importance of long term sustainable support to basic services for the research and education communities as well as on the consolidation of global cooperation for Research Data and Computing infrastructures in the above contexts.

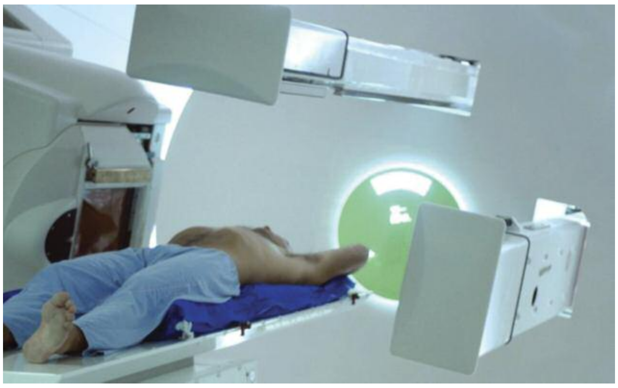
The event is organised with the support of the Italian Ministry of Education, Universities and Research (MIUR), the Italian Supercomputing Center (CINECA), the Italian National Research Council (CNR), the Italian National Institute for Geophysics and Volcanology (INGV) and RDA Europe. High-level policy-makers, national, European, and international scientists, academics, as well as government representatives will be invited to attend.

Participation to this event is by invitation only.

More information:
https://europe.rd-alliance.org/Content/Events.aspx?id=230

World's First Patient Treated by Full 3-D Image Guided Proton Therapy

Proton therapy is considered the most advanced and targeted cancer treatment due to its superior dose distribution and reduced side effects. Protons deposit the majority of their effective energy within a precisely controlled range within a tumor, sparing healthy surrounding tissue. Higher doses can be delivered to the tumor without increasing the risk of side effects and long-term complications, improving patient outcomes and quality of life. The Belgian company IBA is the world leader in the field.



Picture courtesy of IBA, s.a.

View of the proton therapy treatment room.

IBA and the iMagX team at Université catholique de Louvain have jointly developed a software platform and a 3-D cone beam CT in order to guide the proton beam in real time in the treatment room in the frame of a public-private R&D partnership between IBA, UCL and the Walloon Region of Belgium. The system allows for dose delivery estimation and efficient 3-D reconstruction, co-registration of the in-vivo image with the treatment planning based on offline 3-D high resolution Computer Tomography of the patient, both in real time. IBA's AdaPTInsight is the first operational software based on ImagX software. The global system including the hardware and software were granted FDA approval and has been used for the first time to treat a patient in Philadelphia, United States, at Penn Medicine's Department of Radiation Oncology on 9 September 2014. More information can be found at http://www.imagx.org.

Altruism in Game Theory

Research shows that consideration for others does not always lead to the best outcome - that is, when it's applied in game theory. **Bart de Keijzer** (CWI) studied algorithms for game theory, with a focus on cooperative aspects. He defended his PhD thesis 'Externalities and Cooperation in Algorithmic Game Theory' on 16 June at VU University. His research results can have applications in data and traffic networks, peer-to-peer networks and GSP auctions, such as used by Google Adwords.

In conventional models it is a common assumption that players are only interested in themselves. However, in real life players are also influenced by others. De Keijzer investigated the impact of cooperation, friendship and animosity on different games. One of his conclusions is that when players behave altruistic, the flow in a road or data network can become worse. "It's a remarkable result that for the mathematical concept of social welfare, one can sometimes better choose at the expense of others, than to change the strategy to please them," the researcher says.

With these more realistic models researchers and policy makers can make better qualitative predictions. Other research results from De Keijzer, who is now working at the Sapienza University of Rome, have applications in procurement auctions, treasury auctions, spectrum auctions and the allocation of housing. See also http://bart.pakvla.nl/

Essay Contest Prize for Tommaso Bolognesi

In August 2014 **Tommaso Bolognesi**, senior researcher at ISTI-CNR, Pisa, was awarded for the second time (the first was in 2011) a prize in the essay contest "How Should Humanity Steer the Future?", launched by the U.S. institution FQXi (Foundational Questions Institute). His essay, entitled 'Humanity is much more than the sum of humans', ambitiously attempts to use ideas on the computational universe conjecture by Wolfram and others, on life as evolving software and on mathematical biology by G. Chaitin, and on integrated information and consciousness by G. Tononi, for providing some formal foundations to the cosmological visions of the French Jesuit and paleontologist Pierre Teilhard de Chardin. The essay can be found and commented at: http://fqxi.org/community/forum/topic/2014.

ERCIM - the European Research Consortium for Informatics and Mathematics is an organisation dedicated to the advancement of European research and development, in information technology and applied mathematics. Its member institutions aim to foster collaborative work within the European research community and to increase co-operation with European industry.

ERCIM is the European Host of the World Wide Web Consortium.

Consiglio Nazionale delle Ricerche, ISTI-CNR
Area della Ricerca CNR di Pisa,
Via G. Moruzzi 1, 56124 Pisa, Italy
http://www.isti.cnr.it/

Czech Research Consortium
for Informatics and Mathematics
FI MU, Botanicka 68a, CZ-602 00 Brno, Czech Republic
http://www.utia.cas.cz/CRCIM/home.html

Centrum Wiskunde & Informatica
Science Park 123,
NL-1098 XG Amsterdam, The Netherlands
http://www.cwi.nl/

Fonds National de la Recherche
6, rue Antoine de Saint-Exupéry, B.P. 1777
L-1017 Luxembourg-Kirchberg
http://www.fnr.lu/

FWO
Egmontstraat 5
B-1000 Brussels, Belgium
http://www.fwo.be/

F.R.S.-FNRS
rue d'Egmont 5
B-1000 Brussels, Belgium
http://www.fnrs.be/

Foundation for Research and Technology - Hellas
Institute of Computer Science
P.O. Box 1385, GR-71110 Heraklion, Crete, Greece
http://www.ics.forth.gr/

Fraunhofer ICT Group
Anna-Louisa-Karsch-Str. 2
10178 Berlin, Germany
http://www.iuk.fraunhofer.de/

INESC
c/o INESC Porto, Campus da FEUP,
Rua Dr. Roberto Frias, nº 378,
4200-465 Porto, Portugal

Institut National de Recherche en Informatique
et en Automatique
B.P. 105, F-78153 Le Chesnay, France
http://www.inria.fr/

I.S.I. - Industrial Systems Institute
Patras Science Park building
Platani, Patras, Greece, GR-26504
http://www.isi.gr/

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics and Electrical Engineering, N 7491 Trondheim, Norway
http://www.ntnu.no/

SBA Research gGmbH
Favoritenstraße 16, 1040 Wien
http://www.sba-research.org

SICS Swedish ICT
Box 1263,
SE-164 29 Kista, Sweden
http://www.sics.se/

Spanish Research Consortium for Informatics and Mathematics
D3301, Facultad de Informática, Universidad Politécnica de Madrid
28660 Boadilla del Monte, Madrid, Spain,
http://www.sparcim.es/

Science and Technology Facilities Council
Rutherford Appleton Laboratory
Chilton, Didcot, Oxfordshire OX11 0QX, United Kingdom
http://www.scitech.ac.uk/

Magyar Tudományos Akadémia
Számítástechnikai és Automatizálási Kutató Intézet
P.O. Box 63, H-1518 Budapest, Hungary
http://www.sztaki.hu/

University of Cyprus
P.O. Box 20537
1678 Nicosia, Cyprus
http://www.cs.ucy.ac.cy/

University of Geneva
Centre Universitaire d'Informatique
Battelle Bat. A, 7 rte de Drize, CH-1227 Carouge
http://cui.unige.ch

University of Southampton
University Road
Southampton SO17 1BJ, United Kingdom
http://www.southampton.ac.uk/

Universty of Warsaw
Faculty of Mathematics, Informatics and Mechanics
Banacha 2, 02-097 Warsaw, Poland
http://www.mimuw.edu.pl/

Universty of Wroclaw
Institute of Computer Science
Joliot-Curie 15, 50–383 Wroclaw, Poland
http://www.ii.uni.wroc.pl/

Technical Research Centre of Finland
PO Box 1000
FIN-02044 VTT, Finland
http://www.vtt.fi/

Subscribe to ERCIM News and order back copies at http://ercim-news.ercim.eu/