Proceedings of the Projects Showcase @ STAF'15
L'Aquila, Italy. July 22, 2015

Projects Showcase @ STAF'15

Marco Autili
Tom Ritter
Andrey Sadovykh
Massimo Tivoli

Preface

During the last decade a number of national and international financial instruments have been funding research, innovation and development programmes. From local to global scales, many projects have been funded delivering research and industrial innovation through reinforcing the excellence of science, speeding up the industrial leadership, and supporting the growth of new markets.

These initiatives have been driving economic growth and creating employment opportunities, by addressing societal challenges, boosting productivity and securing competitiveness.

Promoting collaborations activities among related projects and boosting the growth of profitable cooperation among the involved partners is an important key enabler to increase the success of these projects, even after their end, and the possibility to conceive new project proposals.

To this aim, the Projects Showcase event at STAF'15 (`http://www.disim.univaq.it/staf2015/projects-showcase/`) is dedicated to national and international projects dissemination and cooperation. The event provides the opportunity for researchers and practitioners (from both academia and industry) involved in completed, to be started or ongoing research projects related to Software Technologies, Applications, and Foundations, to share opinions, propose solutions to open challenges and generally explore novel and emerging ideas for future project proposals possibly advancing existing ones with suitable follow-up projects.

The Projects Showcase received contributions disseminating the objectives of various projects, outcomes of specific deliverables, the project final outcome, as well as, advances beyond the state of the art, overall innovation potential, exploitation approach and (expected) impact, marketing value, barriers and obstacles. Thus, the Projects Showcase at STAF'15 represents a concrete opportunity for participants to share experience, ideas, on-going work, and knowledge that can lead to fruitful collaborations and cross-sectorial concertation among projects.

Many people contributed to the success of the Projects Showcase event at STAF'15. We would like to truly acknowledge the work of all Program Committee members, and reviewers for the timely delivery of reviews and constructive discussions given the very tight review schedule. Finally, we would like to thank the authors, without them the event simply would not exist.

July 22, 2015
L'Aquila, Italy

Marco Autili
Tom Ritter
Andrey Sadovykh
Massimo Tivoli

Organisation

Chairs

Marco Autili	DISIM, University of L'Aquila, Italy
Tom Ritter	Fraunhofer, Germany
Andrey Sadovykh	SOFTEAM, France
Massimo Tivoli	DISIM, University of L'Aquila, Italy

Program Committee

Nikolaos Georgantas	INRIA, Paris – Rocquencourt, France
Rocco Oliveto	University of Molise, Italy
Valérie Issarny	INRIA, Paris – Rocquencourt, France
Paolo Falcarin	University of East London, UK
Luciano Baresi	Politecnico di Milano, Italy
Antonio Brogi	University of Pisa, Italy
Massimiliano Di Penta	University of Sannio, Italy
Apostolos Zarras	University of Ioannina, Greece
Geir Horn	University of Oslo, Norway
Sébastien Gérard	CEA List, France
Holger Giese	HPI, University of Potsdam, Germany
Reda Bendraou	Université Pierre et Marie Curie, France
Alexander Viehl	FZI, Germany

Table of Contents

ARTIST: Model-Based Stairway to the Cloud

Javier Troya[1], Hugo Brunelière[2], Martin Fleck[1], Manuel Wimmer[1], Leire Orue-Echevarria[3], and Jesús Gorroñogoitia[4]

[1] Business Informatics Group, Vienna University of Technology, Austria
{troya,fleck,wimmer}@big.tuwien.ac.at
[2] AtlanMod Team (Inria, Mines Nantes & LINA), Ecole des Mines de Nantes, France
hugo.bruneliere@inria.fr
[3] Tecnalia Research and Innovation, Bilbao, Spain
leire.orue-echevarria@tecnalia.com
[4] ATOS Research and Innovation, Madrid, Spain
jesus.gorronogoitia@atos.net

Abstract. Over the past decade, cloud services emerged as one of the most promising technologies in IT. Since cloud computing allows improving the quality of software and, at the same time, aims at reducing costs of operating software and hardware, more and more software is delivered as a service in the cloud. However, moving existing software applications to the cloud and making them behave as software as a service is still a major challenge. In fact, in addition to technical aspects, business aspects also need to be considered. The ARTIST EU project (FP7) proposes a comprehensive model-based modernization approach, covering both business and technical aspects, to *cloudify* already existing software. In particular, ARTIST employs MDE techniques to automate the reverse engineering and forward engineering phases in a way that modernized software truly benefits from targeted cloud environments. In this paper we describe the overall ARTIST approach and present several lessons learned.

Keywords: ARTIST, Migration, MDE, Cloud

1 Introduction

Dealing with paradigm transitions has always been a recurring problem in software engineering. Nowadays, one of the most popular technological and business trends is to deploy applications on the cloud. This notably allows making pieces of software and related offered services available more easier and dynamically to a wider audience. This also provides some interesting new capabilities, such as improved scalability in contexts where traditional software running in on-premise environments was previously not efficient enough. While the most recent applications may have been designed with cloud deployment in mind, a large majority of the already existing software has been developed in such a way that it is not directly fully cloud-compatible. We refer to this as *legacy* software. As a consequence, there is currently a real need for concrete solutions supporting companies in evolving their legacy applications in order to make them deployable on the cloud, as well as to exploit the full potential and services provided by the cloud.

As an answer to this problem, the ARTIST EU collaborative project [4] aims at facilitating the migration and modernization of legacy software assets and businesses to the cloud. To this intent, it provides a generic customizable model-based methodology and corresponding open source tooling for migrating such applications to the cloud. Covering the traditional reverse engineering and forward engineering phases (i.e., the actual migration), it also addresses (pre-)migration feasibility analysis from both technical and business perspectives as well as (post-)migration verification and certification.

The remainder of the paper is structured as follows. Section 2 gives an overview of the ARTIST project from an general point of view. Then, Section 3 introduces the main objectives of the project and its outcomes. Section 4 details the main innovation aspects brought by ARTIST, and also some encountered obstacles. Section 5 discusses the related work and collaboration with related EU projects. Finally, Section 6 concludes the paper by summarizing the main achievements and presents the ongoing exploitation of the results.

2 Project's Overview

ARTIST, standing for "Advanced software-based seRvice provisioning and migraTIon of legacy SofTware", is an EU Integrated Project (IP) which is part of the Seventh Framework Programme for Research and Technological Development (FP7). It is a still ongoing project that started on October 1, 2012 for a total duration of three years, thus ending in coming September 30, 2015. The project has a total budget of €9,690,258, for a total EC funding of €6,953,705. It directly involves 10 partners coming from 7 different countries which are Spain, France, Germany, Austria, Italy, Greece and Belgium. Academics in the project are coming from internationally recognized institutions: Inria, Fraunhofer, Tecnalia, Vienna University of Technology and Institute of Communication and Computer Systems. Industrial partners vary from innovative SMEs or tool vendors (Sparx Systems, ATC, Spikes) to large service companies (Atos, Engineering). The project is composed of a total of 13 work packages (WPs). WP1-WP4 are organizational WPs, while WP5-WP11 are technical WPs. Finally, WP12 is concerned with the use cases development and WP13 with the tools and methodology evaluation on the use cases. All the information on the ARTIST project is publicly available from the project website[5], including related material as well as access to the ARTIST Open Source Release.

Figure 1 gives an overview of the approach designed and developed in the context of ARTIST. The ARTIST model-based approach covers three main phases:

- **Pre-Migration**. It occurs prior to the actual realization of the migration. It principally consists in ensuring that the migration is feasible and/or desirable, from both a business and technical perspective.
- **Migration**. It can be triggered after the pre-migration phase in case of successful assessment. It is composed of two sub-phases:
  - **Reverse Engineering**. It deals with better understanding the initial application thanks to the (semi-)automated discovery of models describing it as accurately as possible and at different levels of abstraction (from low-level code models up to higher-level design models including, e.g., architectural models).

[5] http://www.artist-project.eu/

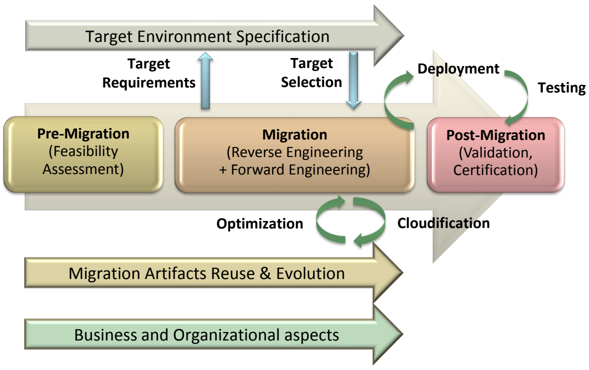
Target Environment Specification
Target Requirements
Target Selection
Deployment
Testing
Pre-Migration (Feasibility Assessment)
Migration (Reverse Engineering + Forward Engineering)
Post-Migration (Validation, Certification)
Optimization
Cloudification
Migration Artifacts Reuse & Evolution
Business and Organizational aspects

Fig. 1. Overview of the ARTIST approach

  - **Forward Engineering**. It reuses the previously obtained models in order to perform the required cloud-oriented adaptations and optimizations onto (parts of) the initial application, with the final objective of sofware (re-)generation.
- **Post-Migration**. It starts after the actual migration has been fully performed in order to evaluate the resulting cloud-compatible pieces of software. It notably consists in validating that the migrated application behaves similarly to the original one once deployed to the cloud, and certifying the compliance of the migrated solution to common cloud practices.

In parallel to these activities, ARTIST also provides support for better handling the cloud target environment identification and selection process (if relevant, in some cases the target cloud platform could be imposed for various reasons). Complementarily, it comes with a dedicated repository for storing and retrieving useful (modeling) artifacts produced in past migration projects (e.g., common metamodels, generic model transformations or skeletons of extensible transformations, code generators, etc.).

3 Objectives and Expected Outcomes

ARTIST provides both an overall model-based methodology and corresponding open source tooling to apply it on real migration scenarios. These ARTIST outcomes are fully reflected within the official ARTIST Open Source Release[6], which gives free public access to the project results in a transparent manner.

Thus, relying on the model-based approach shortly summarized in Section 2, ARTIST provides a set of concrete outcomes that can be practically used and deployed in the context of migration-to-the-cloud projects. Figure 2 summarizes these main outcomes.For a more detailed description and related references (e.g. already published articles), see Section 4.

The first ARTIST asset is its generic and customizable methodology that comes under the form of 1) a complete handbook literally describing it and 2) a Methodology

[6] http://www.artist-project.eu/open-source-package

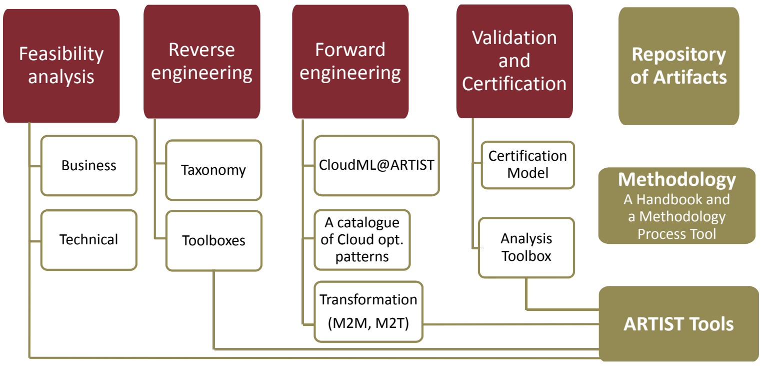

Fig. 2. The main ARTIST outcomes

Process Tool (MPT) allowing to plan, model and follow its underlying processes within the context of various migration projects. In order to practically implement this methodology, ARTIST offers the so-called ARTIST Tools as a second main asset. Interestingly, a large majority of its components is part of the ARTIST Open Source Release.

This ARTIST Tools asset is currently composed of individual components or tool sets covering four fundamental phases of the ARTIST approach. The pre-migration activity of analyzing a given migration feasibility prior to its actual realization can be made thanks to the combined use of the Maturity Assessment Tool (MAT), Business Feasibility Tool (BFT) and Technical Feasibility Tool (TFT). To support Model-Driven Reverse Engineering (MDRE), ARTIST proposes a taxonomy of legacy artifacts guiding engineers in the process of doing a very first analysis of the existing application and its internal structure/content. Then two toolboxes, the Model Discovery Toolbox (MDT) and Model Understanding Toolbox (MUT), allow performing initial model discovery and further model understanding activities, respectively. Having obtained all the required models of the original application at the appropriate levels of abstraction, forward engineering can be realized. In this regard, ARTIST provides different valuable artifacts. The first one is the CloudML@ARTIST language, defined as a UML profile, that is intended to support the identification and definition of the cloud target. The second one is a catalogue of common cloud-specific optimization patterns that can be reused (and eventually completed) in the context of various migration-to-the-cloud projects. The last one is a set of already existing model transformations and code generators that are either completely generic or to be customized to particular migration scenarios. Finally, the migrated software is analyzed to see if the behavior of the legacy software is preserved and if the non-functional requirements are compliant. A certification of the migration (e.g., as a consultancy service) can also be realized.

The ARTIST Repository is another significant outcome of the project. It provides a centralized way of dealing with all the reusable artifacts that can be relevant in such a migration context (cf. the reusable modeling artifacts above-mentioned for instance).

An empirical evaluation of the methodology and the tools is currently being conducted by the use case providers during this last year of the project [4].

4 Main Innovations and Encountered Obstacles

Even though all tasks and work packages are inter-related and their outcomes are used together in the overall ARTIST solution, each one investigates its own line of research. For this reason, the ARTIST project has advanced the state-of-the-art in different fields, unavoidably facing some obstacles and barriers. In this section we present the advances realized in the context of the tasks involving the phases presented in Section 2, and express the difficulties we have found. In particular, we cite some of our works from the more than 30 research papers already published at the time of writing.

4.1 Advances in Different Fields in the Main Three Phases

The *Modernization assessment* task covers the pre-migration phase mentioned in Section 2. In [1, 19, 20], we presented an innovative analysis combining technical and business dimensions in order to assess the maturity of an application and the convenience of migrating it to the cloud. It is based on quantitative indicators always ensuring the company's business continuity. In [2], we conducted a practical application of this pre-migration phase in a particular scenario.

Task *Legacy Product Analysis by Reverse Engineering* corresponds to the reverse engineering phase of the migration process (cf. Section 2). In [10], we presented the MoDisco open source MDRE framework which is used as our overall approach for both discovering initial models from the system artifacts and further understanding them. In the context of ARTIST, we especially advanced on the *model discovery* of behavioral UML2 Activity models from source code, whose implementation is still ongoing. In [5, 6] we presented JUMP, a framework which offers the possibility to discover UML profiles out of annotations at code level and which we integrated within the MoDisco approach. This framework advances the field by finding an effective mapping between Java and UML, generating UML profiles from annotation-based libraries. As for the *model understanding* phase, in [4] we presented an approach to obtain only the parts of a metamodel that we are interested in. This is realized by a type-safe restructuring of snippets that are generated from base metamodels. Also, in [12], we presented an automatic approach to obtain a component model from a class diagram using search-based optimization techniques.

The *New software generation by forward engineering* task represents the forward engineering phase. In order to optimize the application with regards to its non-functional properties, we presented in [12, 14] a search-based software engineering (SBSE) approach to select the proper set of optimization patterns, out of a catalogue of these patterns (cf. Section 3), to apply. In such works, we proposed to optimize the model of the application through in-place transformations. Thus we presented the MOMoT framework which provides several algorithms for local and global searches of rule applications guided by single and multiple objectives formulated in terms of models. We presented in [7] how CloudML@ARTIST facilitates expressing cloud-based deployments directly in UML, which is especially beneficial for migration scenarios where reverse-engineered UML models are tailored towards a selected cloud environment. Since the process of forward engineering is driven by model transformations, we proposed in [8] the concept of *patch transformations*, as created in the context of co-evolution, where only the part that has changed needs to be re-executed in a model transformation.

Task *Migrated product testing, validation and certification* represents the post-migration phase, where the quality of the modernized software is to be evaluated. In [13], we studied how FUML can be used to study the non-functional properties of UML models, without needing to translate the latter into any other formalism. We presented in [18] an approach for integrating existing software libraries with FUML models, so that they can be considered in model simulation. Also aiming at comparing models for the legacy and cloud versions of the applications, we propose in [16] a generic semantic differencing approach that can be instantiated to realize semantic diff operators for specific modeling languages. Finally, we investigated on how to find bugs in model transformations [11], which play a central role in the forward engineering process. We came up with an approach to systematically mutate model transformations [21], which is an important step in the process of identifying bugs.

4.2 Encountered Obstacles

The heterogeneity of programming languages and frameworks present in already existing software, as well as the plethora of existing platforms for deploying applications to the cloud, have been the main obstacles that we have found in our project. The purpose of the ARTIST project is not to define a specific migration process for a particular technology, but rather to propose a semi-automatic generic software migration approach. However, even if the approach is generic, we still need to rely on some technical components that can be quite specific. Furthermore, coming up with such a generic approach is not easy when actual inputs can vary significantly from some applications to others.

For these reasons, we pragmatically dediced to focus on particular technologies for instantiating specific parts of our generic process. For instance, we have been able to obtain UML profiles from Java libraries. We would need to slightly modify the implementation in order to also consider C# code. As another example, for the generation of code from UML models, we have focused on Java and C# as key languages for our industrial partners. However, other target programming languages may also be considered in the future.

In any case, these limitations are precisely the reason why we decide to follow a model-based approach. Raising up the level of abstraction allows us to reason about the software application properties in a platform-independent manner, and consequently to reason about adaptations for the cloud without being polluted by too low-level technical aspects. These adaptations are therefore abstracted away from any technology and can be generically (re)applied in different scenarios.

5 Related Work and Projects

Since cloud computing is a relatively novel computing paradigm, several ongoing research and European projects are currently dealing with the many different issues regarding cloud systems modeling. For instance, MODAClouds [3] and PaaSage [15] also propose, among their objectives, some model-based migration support. Currently, we are collaborating with MODAClouds and PaaSage to come up with a common modeling language for cloud software by merging the languages that have been created in the three projects.

The SeaClouds project [9] takes care of different aspects of the cloud development life-cycle, such as an open, generic and interoperable foundation to orchestrate parts of cloud-based applications. It provides services to monitor, manage and migrate the underlying providers (both public and private clouds) and thus leverages SLA policies in order to guarantee the required performance and QoS on multi-cloud environments. We have already collaborated with this European project in the study of the analysis of non-functional properties of systems [17].

6 Conclusions and Ongoing and Future Exploitation

The ongoing ARTIST project intends to provide relevant support for making easier the process of migrating already existing (legacy) applications to the cloud. It is currently resulting in a general model-based methodology and corresponding open source tooling allowing to implement it in the context of real industrial migration projects.

We are now in the process of preparing the project's follow-up in terms of further exploitation of relevant results, both for the ARTIST consortium as a whole and from an individual partner perspective. As already mentioned, ARTIST has a committed open source exploitation strategy where tools are built by contributors to the open source community. At the same time, the commercial partners in the project have high aspirations for the results. There is a strong potential for partnerships based on geographic coverage, skills and IPR synergies as well as links to the partner's existing portfolios. The partners have been thinking about a way to formalize collaboration in exploitation, such as pooling investment in development, marketing and cross fertilization of opportunities (respecting the open source distribution of software and their license terms).

Consequently, the consortium plans to form a so-called "ARTIST Club" based on a legal agreement which controls the use and ownership of branding associated with the project. In this way, the software can be used under the terms of the license while organizations cannot market services based on them under the ARTIST brand. Ultimately, then, the ARTIST Club serves as a marketing umbrella through which a greater presence can be reached for less resources than through diluted individual marketing investments. At the time of writing, the ARTIST Club contract is being drafted and project participants are finalizing exploitation plans based on opportunities through it.

Acknowledgement

This work is co-funded by the European Commission under the ICT Policy Support Programme, grant no. 317859 (ARTIST project).

References

1. Alonso, J., Orue-Echevarria, L., Corera, Z., Gorronogoitia, J., Karaboga, B.: ARTIST Technical Feasibility Tool: Supporting the early technical feasibility assessment of application cloudifications. In: Proc. of ICSEA (2014)

2. Alonso, J., Orue-Echevarria, L., Escalante, M., Gorronogoitia, J., Presenza, D.: Cloud modernization assessment framework: Analyzing the impact of a potential migration to Cloud. In: Proc. of MESOCA (2013)
3. Ardagna, D., di Nitto, E., Mohagheghi, P., Mosser, S., Ballagny, C., D'Andria, F., Casale, G., Matthews, P., Nechifor, C.S., Petcu, D., Gericke, A., Sheridan, C.: MODAClouds: A model-driven approach for the design and execution of applications on multiple Clouds. In: MISE@ICSE. IEEE/ACM (2012)
4. Bergmayr, A., Bruneliere, H., Canovas Izquierdo, J., Gorronogoitia, J., Kousiouris, G., Kyriazis, D., Langer, P., Menychtas, A., Orue-Echevarria, L., Pezuela, C., Wimmer, M.: Migrating Legacy Software to the Cloud with ARTIST. In: Proc. of CSMR (2013)
5. Bergmayr, A., Grossniklaus, M., Wimmer, M., Kappel, G.: Bridging java annotations and UML profiles with JUMP. In: Proc. of the Demonstrations Track MoDELS. CEUR Workshop Proceedings, vol. 1255 (2014)
6. Bergmayr, A., Grossniklaus, M., Wimmer, M., Kappel, G.: Jump—from java annotations to uml profiles. In: Proc. of MoDELS, LNCS, vol. 8767. Springer (2014)
7. Bergmayr, A., Troya, J., Neubauer, P., Wimmer, M., Kappel, G.: UML-based Cloud Application Modeling with Libraries, Profiles, and Templates. In: Proc. of CloudMDE@MoDELS. CEUR Workshop Proceedings, vol. 1242 (2014)
8. Bergmayr, A., Troya, J., Wimmer, M.: From out-place transformation evolution to in-place model patching. In: ASE. ACM (2014)
9. Brogi, A., Ibrahim, A., Soldani, J., Carrasco, J., Cubo, J., Pimentel, E., D'Andria, F.: SeaClouds: A European Project on Seamless Management of Multi-cloud Applications. SIGSOFT Softw. Eng. Notes 39(1), 1–4 (2014)
10. Brunelière, H., Cabot, J., Dup´e, G., Madiot, F.: Modisco: A model driven reverse engineering framework. Information and Software Technology 56(8), 1012 – 1032 (2014)
11. Burgueno, L., Troya, J., Wimmer, M., Vallecillo, A.: Static Fault Localization in Model Transformations. IEEE Transactions on Software Engineering 41(5), 490–506 (May 2015)
12. Fleck, M., Troya, J., Wimmer, M.: Marrying Search-based Optimization and Model Transformation Technology. In: Proc. of NasBASE (2015)
13. Fleck, M., Berardinelli, L., Langer, P., Mayerhofer, T., Cortellessa, V.: Resource Contention Analysis of Cloud-based System through fUML-driven Model Execution. In: Proc. of NIM-ALP@MoDELS. CEUR Workshop Proceedings, vol. 1074 (2013)
14. Fleck, M., Troya, J., Langer, P., Wimmer, M.: Towards Pattern-Based Optimization of Cloud Applications. In: Proc. of CloudMDE@MoDELS. CEUR Workshop Proceedings, vol. 1242 (2014)
15. Koller, B.: Model Based Cloud Application Development using PaaSage. Innovatives Supercomputing in Deutschland 11(1) (2013)
16. Langer, P., Mayerhofer, T., Kappel, G.: Semantic Model Differencing Utilizing Behavioral Semantics Specifications. In: Proc. of MoDELS, LNCS, vol. 8767. Springer (2014)
17. Moreno-Delgado, A., Durán, F., Zschaler, S., Troya, J.: Modular DSLs for Flexible Analysis: An e-Motions Reimplementation of Palladio. In: Proc. of ECMFA. LNCS, vol. 8569, pp. 132–147. Springer (2014)
18. Neubauer, P., Mayerhofer, T., Kappel, G.: Towards Integrating Modeling and Programming Languages: The Case of UML and Java. In: Proc. of GEMOC@MoDELS. CEUR Workshop Proceedings, vol. 1236, pp. 23–32 (2014)
19. Orue-Echevarria, L., Alonso, J., Escalante, M., Schuster, S.: Assessing the Readiness to Move into the Cloud. In: Cloud Computing. Springer (2013)
20. Orue-Echevarria, L., Escalante, M., Alonso, J.: An Assessment Tool to Prepare the Leap to the Cloud. In: Cloud Computing. Springer (2013)
21. Troya, J., Bergmayr, A., Burgueño, L., Wimmer, M.: Towards Systematic Mutations for and with ATL Model Transformations. In: Mutation Workshop @ ICST (2015)

Learn PAd : Collaborative and Model-based Learning in Public Administrations

Antonia Bertolino[1], Guglielmo De Angelis[1,4], Andrea Polini[2], and Darius Šilingas[3]

[1] CNR–ISTI, Italy
{antonia.bertolino,guglielmo.deangelis}@isti.cnr.it

[2] University of Camerino, Italy
andrea.polini@unicam.it

[3] No Magic Europe, Lithuania
darius.silingas@nomagic.com

[4] CNR–IASI, Italy

Abstract. In modern society public administrations (PAs) are undergoing a transformation of their perceived role from controllers to proactive service providers. PAs are today under pressure to constantly improve the quality of delivered services, while coping with quickly changing context (changes in law and regulations, societal globalization, fast technology evolution) and decreasing budgets. As a result civil servants delivering such services to citizens are challenged to understand and put in action latest procedures and rules within tight time constraints. The European project Learn PAd copes with this transformation by proposing an e-learning platform that enables process-driven learning and fosters cooperation and knowledge-sharing. The platform supports both an informative learning approach, based on enriched business process (BP) models, and a procedural learning approach, based on simulation and monitoring, while relating them as well to learning objectives and key performance indicators.

1 Introduction

The Public Administration (PA) sector in modern society is characterized by the need to continuously improve the delivery of already provided services as well as to introduce new services for both citizens and companies. The adoption of the European Interoperability Framework [1] challenges the PAs from the European member states to cope with several and interconnected procedures, which are often documented and modelled in term of Business Processes (BPs).

A BP describes a collection of activities, messages, and forms that the PAs have to elaborate in order to produce a service to their end-users (i.e citizen, company, or other PAs). Such services are usually articulated in three main phases: i) activation (i.e. request, documentation); ii) processing while performing activities that add value (i.e., checks), or producing evidences: in both these cases the activities use resources (i.e., humans, information, structures); iii) release of a set of produced artefacts as output (e.g. permission, licenses, or rights). Furthermore, it is evident that a BP of a PA must also comply with all regulations governing the subject of the service.

Designing and understanding BPs for PAs is a complex, interdisciplinary, and time consuming task that often involves judgements based on domain knowledge and experience. From the point of view of a civil servant, complexity is also raised by the fact that BPs typically include several alternative scenarios, many of which are seldom activated and may be thus not well understood. Moreover in most of the cases a collaborative effort involving several PA offices is required to fulfill BP objectives. Finally the introduction of new regulations, or their frequent modifications, results in the intertwined modification/deletion of already established BPs, or it can lead to the creation of BPs that were not originally considered. In short BPs in PAs require that civil servants acquire a complex knowledge [2].

Such knowledge is necessary both for accomplishing the required tasks, and to determine what are the tasks to be executed in presence of uncertainty. The management of such a knowledge is challenging per-se. Civil servants deal with heterogeneous information usually learned from previous engagements. In some cases, they can access to insights from prior projects, where notes for subsequent process steps are scattered among manifold "knowledge containers" spanning from the personal memory/notes, to some official information systems. Nevertheless, it is often difficult to use such pieces of "best practice" in a coordinated manner that can take into account both the documents content and the document context (i.e. the creation situation, the potential usage situation). Therefore civil servants are never done with learning how to carry out their tasks. This is one of the major critical issues that modern PAs have to cope with when transforming their regulation framework in order to improve efficiency and effectiveness.

The Learn PAd project (see Section 5 for details) investigates means for merging the activities modeled in a BP and the context behind, so to collaboratively organize knowledge archives that could support civil servants in learning and mastering the complexity of PA processes. In particular, we promote the engagement of civil servants in learning activities at different times, following different paradigms, and by different means, i.e., by fostering informative learning paradigm (i.e. from the BP models and their related material), but also by promoting the same civil servants as "prosumers" (providers/consumers) of the learning materials.

The rest of the paper is organized as follows: Section 2 envisions a methodology supporting the definition of both BP models, and their associated learning artefacts in PAs. Section 3 presents the overall strategy followed in the project work-plan. Section 4 discusses both the challenges, and the perspectives expected by the Learn PAd project; finally Section 5 concludes the paper by reporting a synthetic outline of the project, the founding schema, the consortium, and the key-people involved.

2 Models and Contents Production Process

A service offered by a PA can be fully carried out within an administration, or may require information held by other institutions, or may be delivered in collaboration with other PAs. In any of these cases the activities that a PA has to perform are strictly guided by rules emphasizing the recognition of the legality and regularity of the administrative work. In addition to the formal obligations, an important characteristic in PA processes is the presence of internal procedures resulting from the establishment of "habits": for example checks or obligations that in the past were required by the official regulations and were afterwards revoked, but are still locally enforced. These internal procedures are frequently part of tacit knowledge of a PA. They often result elusive to highlight and to learn when they are benign, but they are particularly difficult to eradicate when they are mere bureaucracy. In any of these cases, they make the work of new employees difficult. The possibility to collaboratively model and discuss PA processes, in order to document them, can help the training of new employees, and can facilitate the reorganization of offices and the removal of unnecessary procedures.

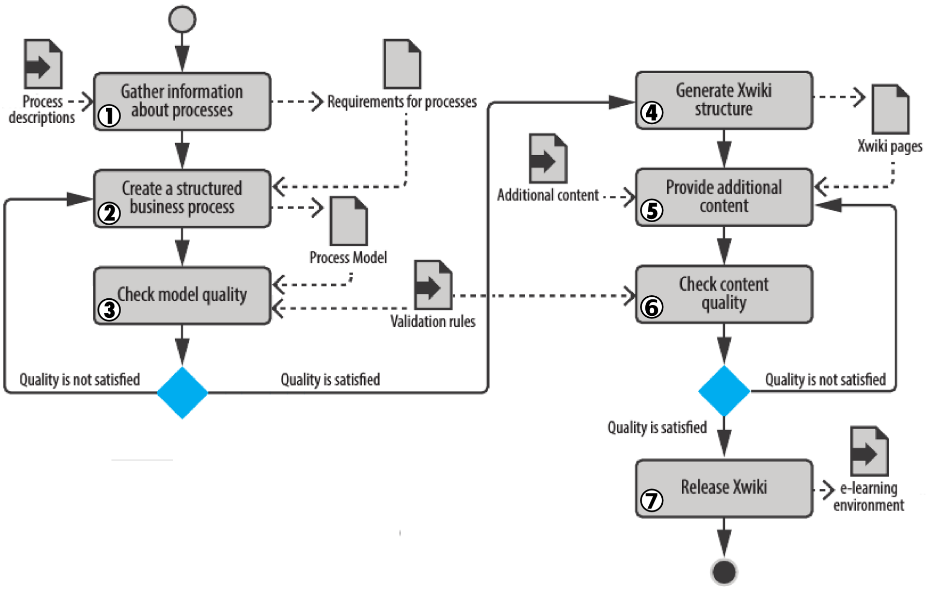

Fig. 1. Models and Contents Production Process

The Learn PAd project envisions a methodology for the collaborative modelling and learning of BP-oriented procedures in PAs. Figure 1 sketches the main steps about the models and contents production process. According to this vision, the organization of a set of activities in a PA starts with the identification of the formal set of laws, rules and procedures regulating them (①). All this information is explicitly structured and codified using a graphical BP notation (②) and is possibly linked with other artefacts modelling information objects (e.g. documents and data), organizational structure business motivation (e.g. policies, strategy, goals, objectives and influencers) [3].The models and their relations are checked in order to assess their structural and semantic quality (③); if these checks fail, further iterations on the modelling phase are requested, else the models are the input of an automatic synthesis process that structures and populates the e-learning platform supporting wiki-like facilities (i.e. the Learn PAd platform – ④). Note that most information about a PA process comes from sets of regulations in natural language (e.g. laws, and procedures that cannot be modified). Thus often quality issues on the model representation (e.g. non termination of the BP, or deadlock) are due to alternative interpretations of such regulations.

The models structured as a collection of interlinked wiki pages provide a valuable source of information grasping the BP objectives and context. In this sense, the Learn PAd platform enables informative and process-driven learning, as the civil servants can browse and access the information codified in the models. The wiki-like representation of the BP structure, its data, and its business rules, enable communities of civil servants to collaboratively develop documentations, hints, people to contact (⑤). Moreover it fosters the continuous upgrade of such information directly by the personnel working within each BP. In other words, the platform fosters cooperation and knowledge-sharing among civil servants. For each new contribution, further quality procedures and check about the contents produced are enabled (⑥); for example in order to support the identification of potential ambiguities or contradictory documentations. If no issues are raised, contents are released on the platform becoming part of the additional documentation of a process (⑦).

In some case, the collaborative development of documentations could reveal unexpected behaviour or desirable improvements. There, it is possible to take advantage of the semi-formal representation of the BP. Specifically, modellers can be notified with a bottom-up feedback about the highlighted problem; when a fresh version of the global specification is released, the related documentations can be preserved by navigating and comparing the two BP versions.

3 Overall Strategy and Work Plan

Main aim of the project is to build a collaborative and BP-based e-learning platform targeted at supporting training of civil servants. Following this objective, the Learn PAd consortium identified five major solution dimensions: *Model Based Learning* where a set of model kinds, and their interrelations, are the key abstractions used in order to represents the different aspects of an organization; *Collaborative Content Management* referred to a collection of wiki pages organized in accordance to the structure of corresponding models; *Content and Models Automatic Assessment* for both the models and their associated documentation produced by civil servants; *Simulation Based Learning Support* that enables civil servants to experiment collaboratively their acquired knowledge; *Monitoring and KPI for engaging learners* where indicators are derived and monitored in order to assess both civil servants, and organizational learning.

These solution dimensions are investigated within four work packages (i.e. WP3, WP4, WP5, WP6) mainly conceived to enhance the research with the area of technology-enhanced learning. These work packages leverage on the activities from other three work packages (i.e. WP1, WP2, WP7) enabling the engineering of the Learn PAd platform. The project validates its achievements using two demonstrators (i.e. WP8). Finally, dissemination and management activities are the focus of the remaining WPs (i.e. WP9, WP10).

In the following we report more details for each work package in Learn PAd.

WP1: Requirements Analysis is conceived as the very first interaction point of the project where all partners meet in order to define and to elicit their needs and their vision for the e-learning platform that will result from the project. Specifically, the whole consortium discusses the requirements from research, methodological, technical, architectural and user perspectives.

WP2: Learn PAd Platform designs the high level architecture for the whole Learn PAd platform and it shapes the strategy for the integration of the various components. It also provides the design of the core functionalities of the collaborative framework to which all the tools developed by WP4, WP5, WP6 can be plugged in. Moreover the core of the collaborative framework relies on the data metamodel resulting from WP3 and WP5.

WP3: Approaches Enabling Model-Based Learning analyses, specifies, and defines the domain-specific models and meta-models that are needed to be adopted for the management and learning of business processes and their context. Specifically, it investigates on adequate means for automatically relating BP-notations such as BPMN [4], and other model kinds representing either the organization of the PAs or learning aspects.

WP4: Models and Contents Quality Assessment investigates on the quality assessment of the BP model specifications, and its related learning contents. The quality assessment will be based on both formal verification, and natural languages processing techniques. The former will focus on proper approaches assessing the satisfaction of relevant properties by defined BP models. The latter will include mechanisms to check if the text within the learning content referring to the elements of a model is *consistent* (e.g. not contradictory, issuing potential ambiguities, etc.) with respect to the model itself.

WP5: Collaborative Content Management deals with the collaborative management of learning contents, including their production and fruition. Specifically the WP is designed around the definition of ontologies providing a shared understanding of learners, contents and KPI. It also leverages on automatic reasoning in order to measure the success of learning, as well as cooperation and content production. This work package fosters the development of a performance culture by explicitly referring to KPIs as means for assessing how learners engage in learning, and improve their competence by means of the platform.

WP6: Simulation Based Learning researches on the design and the construction of a simulation environment where learners are engaged in training activities. The simulation engine will support both the software emulation of the involved parties, and the provisioning of dedicated means for gathering learners willing to train on the business process by interacting with other learners.

WP7: Coordination Activities Supporting Integration is a technical work package prescribing the guidelines for the development and the integration of the software provided by the other work packages.

WP8: Demonstrators aims at demonstrating applicability, acceptance and effectiveness of the overall Learn PAd platform. The WP leverages on the results from the WPs 1-6 from two different PAs, each one providing and highlighting different aspects.

The first case concerns the administrative procedures related to the participation to a European research project. This case is demonstrated in the administrative departments at University of Camerino. It is conceived to engage different partners in the definition of BP models and BP documentation, but without crossing the border of that PA.

The second case addresses a more complex inter-organizational scenario involving several PAs. Specifically, it refers to the activities that the Italian PAs have to put in place in order to allow entrepreneurs to set up a new company (i.e. Sportello Unico Attività Produttive – SUAP). This demonstrator is run under the support of Regione Marche PA, which partially regulates the activities in SUAP, and it also includes a stable school for PA employees.

WP9: Dissemination and Exploitation deals with managing all activities including dissemination, exploitation, standardization undertaken throughout the project.

WP10: Management includes all management tasks (e.g. budgeting, project staffing, planning, quality assurance, etc.).

4 Challenges and Perspectives

For each of the five main solution dimensions presented in Section 3, the Learn PAd project deals with challenges and perspectives overcoming the current state-of-the-art in technology-enhanced learning.

Model Based Learning : The more evident perspective promoted by the Learn PAd project is to adopt models (and their graphical representation) as the basis for organizing knowledge and training people in PAs. In fact models have been used more and more to represent different dimensions of an organization; what Learn PAd proposes is a systematic way to automatically derive and organize e-learning artefacts from such models; thus a platform for Model Based Learning supporting off-line learning by consulting and commenting contents related to specific topic, and learning-by-doing by simulating the BP activities.

A challenge with respect to the current state-of-the-art in the field of Model-Driven Engineering concerns the proposition of means for the homogeneous linking of orthogonal aspects. For instance from the BP notations towards several other model kinds that are referred in the domain of PA (i.e. case, organization, competency, business motivation, data). Only an explicit formulation of those interrelations can enable coherent cross-model analysis, and the automatic synthesis of e-learning artefacts.

This solution dimension is mainly investigated in WP3, and WP5.

Collaborative Content Management : The main innovation brought to the Collaborative Content Management dimension in e-learning is to have an asynchronous learning platform where civil servants are both learners and instructors and contribute to create a shared knowledge. Like in a wiki, the platform supports the provision from some power-brokers of tutorial/learning material that is updated and commented by civil servants in a collaborative way. As introduced above the underlying structure of the platform is automatically derived according to precise metamodel specifications and reflects the structure of different model kinds adopted and their interrelations.

In addition, the platform refers to complex ontologies in order to classify learners and contents, and also in order to infer learning needs for the civil servants. Indeed, the innovation about this dimension also concerns the possibility to provide context-based recommendation of both contents and social profiles (e.g. a set of competent colleagues which may provide help).

This solution dimension is mainly investigated in WP5 and partially in WP3.

Content and Models Automatic Assessment : The platform is including automatic verification techniques with reference to both stored models and contents. The research along this dimension leverages formal verification techniques that are used to highlight potential issues in BP and related specifications. Since Learn PAd encourages collaborative work on documenting models on-line with new learning artefacts, the perspective of an automatic support for contents production assessment is particularly appealing. Instead of pre-defined structured information, unstructured information in natural language that is related to the models (comments, annotations, etc. in the platform) need to be analysed. The challenge is that quality for the contents produced by the civil servants is addressed by investigating advanced Natural Language Processing techniques [5]. In this respect, textual analysis mechanisms rely on the definition of rigorous specification (i.e. metamodels and weaving models) for relating the models and with contributed textual material in order to highlight possible inconsistencies or mismatches.

This solution dimension is investigated in WP4.

Simulation Based Learning Support : Another perspective pursued by the Learn PAd project focuses on highly interactive media-intensive learning environments. The challenge is to enhance traditional training methods by means of simulation exercises. In addition, the simulation environment of Learn PAd includes a BP execution engine supporting the behavioural emulation of the involved parties in the process. Also, in this context the simulation may suggest some “interesting” paths in a BP, leveraging research in model-based testing approaches enabling the detection of critical branches/paths [6][7], or seldom activated during actual BP engagements (i.e. paths where civil servants are supposed to be less skilled).

Along this dimension, Learn PAd investigates topics on collaborative simulations, and *gamification*, which are currently considered as promising approaches to learning by many communities in the field [8][9][10].

This solution dimension is investigated in WP6.

Monitoring and KPI for engaging learners : The Learn PAd project envisions KPIs as specifically tailored to reveal knowledge resources and specific capabilities of learners. For example, a perspective foresees individual KPI profile as a mean for learners to identify and to communicate with relevant peers/experts in order to explore collaboration opportunities and enhance their learning process.

The challenge on the KPI dimension rises from their formulation and usage. The research in Learn PAd aims at combining the knowledge maturing scorecard approach which measure the achievement of learning goals, with the Business Motivation Model that explicitly represents the goals and objectives of an organization (i.e. the PA). Instances of the individual KPI profile rely on data from the activities performed within the platform: either with respect to the quantity/quality of learning artefacts produced, or with respect to the results from the sessions executed in the simulation environment.

The Learn PAd project investigates the track of activities on the platform by means of multi-source monitoring facilities grounded on research about the Events-Driven software architecture pattern[11]. In this respect the challenge is to conceive adaptive approaches that are able to correlate auditing data and inferring complex events so that to be used in order to compute the aggregated data from an instance of an individual KPI profile.

This solution dimension is manly investigated in both WP5, and WP6.

5 Learn PAd at a Glance

Project Acronym: Learn PAd
Project Name: Model-Based Social Learning for Public Administrations
Official Project Web-Site: http://www.learnpad.eu
Source of funding: European Commission : EU FP7-ICT-2013-11 / 619583
Overall Total Budget: Total cost: € 3.532.993 (EC contribution: € 2.635.000)
Duration: 30 months (from the 1^{st} Feb. 2014 to the 31^{st} Jul. 2016)
Consortium:

Beneficiary Name	Short Name	Country
Consiglio Nazionale delle Ricerche (coordinator)	CNR	Italy
BOC Asset Management GmbH	BOC	Austria
LINAGORA GSO	LIN	France
No Magic Europe	NME	Lithuania
Regione Marche	MAR	Italy
University of Applied Sciences and Arts Northwestern Switzerland	FHNW	Switzerland
University of Camerino	UNICAM	Italy
University of L'Aquila	UDA	Italy
XWiki SAS	XWIKI	France

Key People:

Project Coordinator : Antonia Bertolino (CNR)
Scientific Leader : Andrea Polini (UNICAM) **Exploitation Leader :** Darius Šilingas (NME)
Technical Leader : Guglielmo De Angelis (CNR) **Project Secretary :** Daniela Mulas (CNR)
WPs Leaders : Robert Woitsch (WP1 – BOC), Jean Simard (WP2, WP7 – XWIKI), Alfonso Pierantonio (WP3 – UDA), Stefania Gnesi (WP4 – CNR), Barbara Thönssen (WP5 – FHNW), Jean-Pierre Lorré (WP6 – LIN), Barbara Re (WP8 – UNICAM), Darius Šilingas (WP9 – NME), Antonia Bertolino (WP10 – CNR)

References

1. European Commission: European Interoperability Framework (EIF) Towards Interoperability for European Public Services (2011)
2. Di Ciccio, C., Marrella, A., Russo, A.: Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. Journal on Data Semantics **4**(1) (2015) 29–57
3. Karagiannis, D., Hrgovcic, V., Woitsch, R.: Model driven design for e-applications: the meta model approach. In: Proc. of 13th International Conference on Information Integration and Web-based Applications and Services, Ho Chi Minh City, Vietnam, ACM (2011) 451–454
4. The Object Management Group: Business Process Modeling Notation (BPMN) version 2.0. (2011)
5. Ferrari, A., Lipari, G., Gnesi, S., Spagnolo, G.O.: Pragmatic ambiguity detection in natural language requirements. In: Proc. of 1st International Workshop on Artificial Intelligence for Requirements Engineering, Karlskrona, Sweden, IEEE (2014) 1–8
6. De Angelis, F., Fanì, D., Polini, A.: Partes: A test generation strategy for choreography participants. In: Proc.of 8th International Workshop on Automation of Software Test, San Francisco, CA, USA, IEEE (2013) 26–32
7. De Angelis, F., De Angelis, G., Polini, A.: A counter-example testing approach for orchestrated services. In: Proc. of the 3rd International Conference on Software Testing, Verification and Validation, Paris, France, IEEE CS (2010) 373–382
8. Alexandrov, N., Velarde, R.R., Alexandrov, V.: Technological Advances in Interactive Collaborative Learning. Chapman & Hall/CRC (2012)
9. Curtis, D.D., Lawson, M.J.: Exploring collaborative online learning. Journal of Asynchronous learning networks **5** (2001) 21–34
10. Monetta, G., Orciuoli, F., Santoro, G.: Collaborative learning and knowledge management: A case study. In: Driving the Economy through Innovation and Entrepreneurship. Springer (2013) 429–441
11. Ben Hamida, A., Bertolino, A., Calabrò, A., De Angelis, G., Lago, N., Lesbegueries, J.: Monitoring service choreographies from multiple sources. In: Proc. of the 4th International Workshop on Software Engineering for Resilient Systems. Volume 7527 of LNCS., Springer (2012) 134–149

OCCIware - A Formal and Tooled Framework for Managing Everything as a Service

Jean Parpaillon[1], Philippe Merle[2], Olivier Barais[1], Marc Dutoo[3], and Fawaz Paraiso[2]

[1] Inria Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France
[2] Inria Lille - Nord Europe, 40 Avenue Halley, 59650 Villeneuve d'Ascq, France
[3] Open Wide, 151 Boulevard Stalingrad, 69100 Villeurbanne, France

Abstract. The OCCIware project aims at building a comprehensive, coherent while modular model-driven toolchain for managing any kinds of cloud resources, especially Data Center as a Service, Deployment as a Service, Big Data as a Service, and Linked Open Data as a Service. Leveraging the Open Cloud Computing Interface (OCCI) and its core model, the OCCIware toolchain applies a model-driven engineering approach based on a formal model of cloud resources and systems. This approach allows for better modularity, clear separation between functional (cloud resources) and non-functional concerns (security, scalability, reliability, etc.). The project brings together ten French partners - academics, SMEs, associations - and is supervised by a Strategic Orientation Committee of eleven top industrial and academic experts. The OCCIware project has been selected by French Ministry of Industry and funded by French Banque Publique d'Investissement (BPI).

1 Project Facts and Figures

Name: OCCIware
Call: French Ministry of Industry, "Investissement d'Avenir" (*Investment for Future*) programme, 4th "Cloud Computing and Big Data" call.
Source of funding: French "Banque Publique d'Investissement" (*Public Bank for Investment*).
Amount of funding: 3,3 M€.
Overall total budget: 5,6 M€.
Man power: 72 *men/year*.
Project consortium: 10 partners (see Table 1).
Involved people: `http://www.occiware.org/bin/view/About/Contributors`
Project Website: `http://www.occiware.org`
Project duration: 36 months.
Current Status: Started from December 2014.
Standardization Impact: supported by Open Grid Forum (OGF) standard development organization [6].

Support: Five French competitive clusters supporting the project (Systematic, Minalogic, PICOM, Images & Réseaux, Solutions Communicantes Sécurisées).

Strategic Orientation Committee: The complete list of members is available at `http://www.occiware.org/bin/view/About/Strategic_Orientation_Committee`

Open Source Software Resources: `https://github.com/occiware`

#	**Name**	**Type**	**Work Package (bold: lead)**
1	Open Wide	SME	**WP1**, all
2	ActiveEon SA	SME	**WP4**, all
4	Institut Mines Télécom / Télécom SudParis	Academic	All
4	Inria	Academic	**WP2**, all
5	Linagora GSO	SME	**WP5**, all but WP3
6	Obeo	SME	**WP3**, all but WP4
7	OW2 Consortium	Association	WP1 and **WP6**
8	Pôle Numérique	Association	All
9	Scalair	SME	All but WP3
10	Université Jospeh Fourier - Grenoble	Academic	All but WP3

Table 1. OCCIware Project Partners

2 Project Overview

2.1 Objectives

While cloud computing has become a reality in most IT domains, migrating existing software to the cloud or developing new innovative added-value cloud resources still require important R&D efforts. Indeed, cloud computing is plagued by heavy partitioning between cloud layers, technical implementations and business domains. For instance, while actual as well as "de facto" market standards have appeared in cloud computing, they are still tied to a particular layer: infrastructure (IaaS), platform (PaaS) or application (SaaS) [22] - actually mostly IaaS, and hardly SaaS.

The aim of the FSN Investissements d'Avenir (Cloud & Big Data 4) OCCIware project is to lower cloud computing adoption costs and break up barriers between its various layers, implementations, domains, by bringing to Open Cloud Computing Interface (OCCI) from Open Grid Forum (OGF) the power of formal methods, model driven engineering (MDE), and Models@run.time, in order to design, model, analyse, simulate, develop, deploy and execute every cloud computing resource as a service.

The OCCIware project aims at providing a comprehensive, coherent while modular model-driven toolchain for managing any kinds of cloud resources, especially Data Center as a Service, Deployment as a Service, Big Data as a Service, and Linked Open Data as a Service. By using a simple resource-oriented meta-model, the OCCIware toolchain will allow to address any kind of resource-based software, drastically reducing development time by using Models@run.time approach [16] and/or code generation while improving overall quality and non-functional aspects of developed software, thanks to the separation of concerns.

Technically, the OCCIware toolchain is extending the Open Cloud Computing Interface from Open Grid Forum, by turning its core model [24] into a formal resource-oriented meta-model and designing new models addressing different domains. It also provides an Eclipse engineering framework for designing, testing and simulating cloud resources. Finally, the OCCIware toolchain includes a generic runtime for executing such designed cloud resources. In its architecture, the runtime implements the separation of concerns allowed by the meta-model, bringing security, reliability and scalability at no cost to developers of cloud resources.

The OCCIware project will be showcased in four demonstrators targetting Data Center as a Service, Deployment as a Service, Big Data as a Service, and Linked Data as a Service. The OCCIware project will be disseminated through Open Source communities (OW2 Consortium, Eclipse Foundation) and standardization bodies (OGF, DMTF) with help from eleven top international industrial and academic experts of the OCCIware's Strategic Orientation Committee.

2.2 Innovations Beyond the State of the Art

Formal methods have been used succesfully in a large variety of domains like processor checking, embedded and critical systems. Aeolus ANR [13] and Mancoosi FP7 [21] projects have delivered the most comprehensive formal model of complex distributed systems. Their project leader is a member of the Strategic Orientation Committee. The OCCIware project aims at describing these models thanks to a single formal meta-model. Nevertheless, to the best of our knowledge, formal methods have not been used in the domain of cloud computing. The OCCIware project aims at proposing **the first formal model for designing and analysing every kind of cloud-based resource-oriented systems**.

This formal model will be based on the first-order relational logic and will be encoded with the Alloy lightweight specification language defined by Pr. Daniel Jackson from MIT [20]. Thanks to Alloy Analyser [14], we will analyse both the OCCIware meta-model and models of cloud resources in order to check their consistency, verify their properties and generate model instances automatically.

Several research projects such as FP7 REMICS [9], FP7 MODAClouds [3], FP7 SeaClouds [10], FP7 PaaSage [8], SINTEF CloudML [1], Eclipse Winery [12], StratusML [11], to cite a few, tackled the provisioning and deployment of multi-cloud applications on existing IaaS and/or PaaS resources through a model-driven engineering approach. These work do not tackle the design and execution of new kinds of cloud resources. Unlike the OCCIware project aims at providing **a model-driven engineering approach to manage every kind of cloud computing resources**.

Several cloud computing standards already exist. The DMTF's Open Virtualization Format (OVF) standard defines a standard packaging format for portable virtual machine images. The DTMF's Cloud Infrastructure Management Interface (CIMI) standard defines a RESTful API for managing IaaS resources only [19]. The OASIS's Cloud Application Management for Platforms (CAMP) standard targets the deployment of cloud applications on top of PaaS resources [4]. The OASIS's Topology and Orchestration Specification for Cloud Applications (TOSCA) standard defines a language to describe and package cloud application artifacts and deploy them on IaaS and PaaS resources [5]. The Eclipse Winery project provides an open source Eclipse-based graphical modelling tool for TOSCA [12] when the OpenTOSCA project provides an open source container for deploying TOSCA-based applications [7] [15]. The FP7 SeaClouds project [10] is based on both OASIS's CAMP and TOSCA standards. The OGF's Open Cloud Computing Interface (OCCI) recommendations [25] propose a generic resource-oriented model [24] for managing any kind of cloud resources, including IaaS, PaaS, and SaaS. Both OVF and OCCI address orthogonal concerns and then are complementary. OCCI is concurrent to CIMI because both address IaaS resource management but OCCI is more general purpose as it can be used also for any kind of PaaS and SaaS resources. CAMP and TOSCA can use OCCI-based IaaS/PaaS resources, so these standards are complementary. The OCCIware project is **based on and extending the OCCI recommendations**.

The FUI CompatibleOne [18] [27] and FP7 Contrail [2] projects have used OCCI recommendations for addressing cloud services interoperability and some partners of the OCCIware project were already involved in these projects. While these two projects have successfully achieved their functional goals, the lack of formal OCCI specifications prevents them to be easily extensible and limits the automation of their implementations. Turning the OCCI core model into a formal meta-model then designing a set of standard models out of it is one objective of the OCCIware project. The OCCIware project will provide **a formal OCCI model supported by a model-driven toolchain facilitating the design, development, and execution of any kind of OCCI-based cloud resources**.

2.3 OCCIware User Story

The OCCIware project outcomes can be illustrated through the following user story. Let a fully resource-oriented application "BeRest". It consumes the following resources through REST web services: compute and storage (IaaS), train and flight timetables (Linked Open Data) and personal calendars (SaaS). It provides the following service as resources: travel booking.

Thanks to our *formal meta-model* and its associated domain-specific language (see Section 3.1), the specifications of resources, including requirements and produced services can be expressed in an homogeneous way and can be verified at design time.

The *OCCIware engineering studio* (see Section 3.2) provides both Eclipse-based *graphical modeler and textual editor* to modelize cloud resources of this application. These tools are then able to expose the application's cloud resources through different points of view, adapted to each actor:

– *architects* for designing the application,
– *developers* for mapping the design onto implementation,
– *CIO* for evaluating overall foreseen infrastructure cost.

Finally, the *OCCIware runtime* (see Section 3.3) is able to execute the application, *ie* mapping resources onto existing services (*e.g.* Amazon EC2 for infrastructure resources) and exposing "BeRest" services as OCCI resources. OCCIware studio tools will be able to configure the runtime for existing features and/or generating extensions through well-known extension points, for handling new features.

3 Project Organization and Outcomes

The OCCIware project has been split up to six work packages:

– **Transversal activities**: Management (WP1) and Communication and Dissemination (WP6),
– **Technical work packages**: Foundations (WP2), Eclipse toolchain (WP3), and Runtime (WP4),
– **Use Cases and Demonstrators** (WP5).

In addition to internal steering committee, a *Strategic Orientation Committee* has been set up to monitor the adequacy of OCCIware strategy with industrial needs and scientific rapidly evolving state-of-the-art.

3.1 WP2 - Foundations

Theorical foundations of the project will produce scientific and formal tools, starting from the OCCI Core Specification. The following outcomes are expected.

The global technical architecture of the project results in a precise description of components and interfaces between the components developed in the project. It is planned to update this document with regard to feedback provided when implementing this architecture.

The OCCI formal model is a formalization of OCCI Core Model. The result is a proven meta-model and a set of constraints on this meta-model. This metamodel will be encoded with Alloy.

An OCCI dedicated language will be developed to express both static and dynamic aspects of the OCCIware models. It may be used for describing resources, manipulate them and simulate interactions between them.

Various OCCI resource models will be developed to address all OCCIware use-case requirements, as well as non-functional aspects of the runtime.

3.2 WP3 - Eclipse Toolchain

The Eclipse-based toolchain must help application developers but also CIOs to embrace the resource-oriented paradigm. The Eclipse Modeling Framework (EMF) is particularly suited for producing this kind of tools. The Obeo partner, as a recognized Eclipse expert and active member of the Eclipse community, will lead these tasks.

First, the OCCI meta-model will be translated into an Ecore meta-model. Eclipse tools will be leveraged to produce a text editor for the dedicated OCCI language implemented on top of Eclipse XText[4]. As the toolchain is dedicated not only to developers but also architects, a graphical modeler will be designed and developed on top of Eclipse Sirius[5]. A model-driven simulator will then be developed on top of CloudSim[6] [17]. The link between the modeling environment and executed applications will be implemented with various code generators or connectors. Generators will generate runtime artifacts like code, configuration files, etc. Connectors will implement the causal link between models and running cloud resources, making OCCIware Models@run.time a reality. Finally, a decision-support tool will be developed to help evaluating the transition from legacy applications to cloud resource-based approach.

3.3 WP4 - Runtime Support

Leveraging the model-driven engineering approach, an execution platform will be able to interpret OCCI models at runtime, providing developers with non-functional aspects in the most transparent and efficient way. While developers can easily model their core business, turning these models into cloud resource-based applications requires a lot more skills due to non-functional aspects: scalable deployment, security, fault-tolerance, etc. Built on top of a kernel able to interpret OCCI models, connectors to existing cloud management interfaces will be developed for monitoring, supervision and distributed deployment. A web-based administration console for OCCI resources is also expected.

3.4 WP5 - Demonstrators and Use-Cases

Four use cases will be developed in the OCCIware project with the objectives of (1) providing requirements to technical work packages, (2) validating the outcomes of the latter and (3) demonstrating the use of the OCCIware toolchain in real industrial environments. The following use cases have been defined:

Datacenter as a Service will demonstrate the use of OCCIware for datacenter management (IaaS) ;

Deploy@OCCIware will offer interoperability layer above existing deployment and monitoring solutions ;

[4] https://eclipse.org/Xtext/

[5] https://eclipse.org/sirius/

[6] http://www.cloudbus.org/cloudsim/

BigData and HPC will use OCCIware to propose scientific applications execution environment as a service ;

LinkedData as a Service will demonstrate the use of OCCIware tools for open linked data based applications.

4 Dissemination and Exploitation

Standardization of methodologies, languages and tools dedicated to resource-oriented software development is the objective of the OCCIware project. Their adoption by targeted audiences will then be a key indicator of the project success. A particular effort has been planned for disseminating technical and scientific results to following targeted communities:

- **Scientific communities** through publications in top journals and conferences. Our precise metamodel for Open Cloud Computing Interface is already published in [23].
- **Industrial communities** will be addressed through industrial events and business clusters.
- All technical outcomes will be published under open source license and then proposed to most appropriate **Open Source communities** (OW2, Eclipse, etc.). Our open source *erocci* generic OCCI Models@run.time is already available at [26].
- Finally, a close relation with OGF **Standards Definition Organization** has been established since the beginning of the project while connections with DMTF and OASIS organizations also exist with the project organization, through OW2 partner and Strategic Orientation Committee members.

Exploitation of the results by partners differs by their really business:

- **Service providers** (Scalair, Pôle Numérique) intend to improve their audience by providing services of high quality, accessible through standard technologies at a lower cost, challenging big actors in their respective market.
- **Integrators and software editors** (Linagora, Open Wide, ActiveEon, Obeo) will benefit from automated development toolchain for integrating at limited cost resource-oriented approach to their existing applications, enabling the access to the huge PaaS and SaaS market.
- **Research institutions** will benefit from the project by establishing their expertise in the first ever formal framework dedicated to the everything-as-a-service paradigm.

5 Conclusion

As the huge majority of software industry is moving toward a fully resource-oriented delivery model, it is time to offer developers a comprehensive toolchain leveraging this convergence to lower development costs and increase the overall quality of resource-oriented applications. The OCCIware project aims at building this toolchain by bringing together existing technological and scientific tools usually promoted in separated communities: formal methods, model-driven engineering, meta-models, Models@run.time, REST architecture style, devops practices. The added value of the project being measurable through its adoption by software developers and scientifics, a particular effort is put to disseminate the OCCI meta-model, models and associated tooling to scientific and industrial communities, but also open source and standardization organizations.

References

1. CloudML Project Website, `http://cloudml.org`
2. Contrail FP7 Project Web Site, `http://contrail-project.eu`
3. MODAClouds FP7 Project Website, `http://www.modaclouds.eu`
4. OASIS CAMP Website, `https://www.oasis-open.org/committees/camp`
5. OASIS TOSCA Website, `https://www.oasis-open.org/committees/tosca`
6. Open Grid Forum Web Site, `http://ogf.org`
7. OpenTOSCA Project Website, `http://www.iaas.uni-stuttgart.de/OpenTOSCA`
8. PaaSage FP7 Project Website, `http://paasage.eu`
9. REMICS FP7 Project Website, `http://www.remics.eu`
10. SeaClouds FP7 Project Website, `http://www.seaclouds-project.eu`
11. StratusML Project Website, `http://www.stargroup.uwaterloo.ca/~mhamdaqa/stratusml`
12. Winery Eclipse Project Website, `https://projects.eclipse.org/projects/soa.winery`
13. Aeolus ANR Project Web Site, `http://aeolus-project.org`
14. Alloy Analyser Web Site, `http://alloy.mit.edu`
15. Binz, T., Breitenbücher, U., Haupt, F., Kopp, O., Leymann, F., Nowak, A., Wagner, S.: OpenTOSCA – A Runtime for TOSCA-based Cloud Applications. In: Service-Oriented Computing, pp. 692–695. Springer (2013)
16. Blair, G., Bencomo, N., France, R.B.: Models@run.time. Computer 42(10) (2009)
17. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 41(1), 23–50 (2011)
18. CompatibleOne FUI Project Web Site, `http://compatibleone.org`
19. Davis, D., Pilz, G.: Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP. vol. DSP-0263 (May 2012)
20. Jackson, D.: Software Abstractions: Logic, Language, and Analysis. MIT press, Revised edn. (2012)
21. Mancoosi FP7 Project Web Site, `http://mancoosi.org`
22. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. National Institute of Standards and Technology 53(6), 50 (2009)
23. Merle, P., Barais, O., Parpaillon, J., Plouzeau, N., Tata, S.: A Precise Metamodel for Open Cloud Computing Interface. In: Proceedings of the 8th IEEE International Conference on Cloud Computing (IEEE CLOUD 2015). IEEE, New York, USA (June 27 - July 2 2015), to appear

24. Nyrén, R., Edmonds, A., Papaspyrou, A., Metsch, T.: Open Cloud Computing Interface – Core. Specification Document GFD-P-R.183, Open Grid Forum, OCCI-WG (Apr 2011), errata update available at http://redmine.ogf.org/projects/occi-wg/repository/show?rev=core-errata
25. OCCI-WG: OCCI Working Group Web Site, http://occi-wg.org/
26. Parpaillon, J.: Erocci Web Site, http:///erocci.ow2.org/
27. Yangui, S., Marshall, I.J., Laisne, J.P., Tata, S.: CompatibleOne: The Open Source Cloud Broker. Journal of Grid Computing 12(1), 1–17 (2013)

Metamodeling Architectures for Business Processess in Organizations

Alfonso Pierantonio[1], Gianni Rosa[1], Darius Silingas[2], Barbara Thönssen[3], and Robert Woitsch[4]

[1] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
Università degli Studi dell'Aquila (Italy)
[2] No Magic Europe
Savanoriu av. 363, LT-49425 Kaunas (Lithuania)
[3] University of Applied Sciences Northwestern Switzerland
Riggenbachstrasse 16, CH-4600 Olten (Switzerland)
[4] BOC Information Technologies Consulting AG
Operngasse 20b, 1040 Wien (Austria)

Abstract. This paper presents the working version of Learn PAd metamodel that captures the concepts that are used to model process-centric business architecture (process, motivation, organization structure, measurements, etc.) in sufficient detail to be useful for on-the-job learning in public administrations. We provides detailed concepts and relationships reusing selected parts of the modeling standards such as BMM, BPMN, and CMMN when possible. The interrelationships between the concepts in different metamodels are captured in a separate weaving model in order to enable encapsulation of separate views that could be specified using isolated standard metamodels.

1 Introduction

In modern society public administrations (PAs) are undergoing a transformation of their perceived role from controllers to proactive service providers. In most cases, the provisioning of the services is a collaborative activity shared among different, possibly many, organisations that are in general quite interrelated. Civil servants are challenged to understand and put in action latest procedures and rules in order to constantly improve their service quality while coping with quickly changing contexts (changes in law and regulations, societal globalization, fast technology evolution) and decreasing budgets.

In order to provide efficient services to citizens and companies, civil servants have to manage and master extremely complex processes in PAs. The Learn PAd[5] project [3] aims at developing a social, collaborative and holistic e-learning platform able to foster cooperation and knowledge-sharing for civil servants. It enables process-driven learning and improvement of the process on a user-friendly basis of wiki pages enriched with additional documentation for a clearer understanding of the process together with

[5] Model-Based Social Learning for Public Administrations is part of the program FP7-ICT-2013.8.2 Technology-enhanced learning. The project started on Feb 1, 2014 and terminate on Jul 31, 2016 with a cost of € 3,535,000. For further detail please refer to http://www.learnpad.eu

guidance based on formalized models. Moreover, the platform supports both an informative learning approach based on enriched Business Process [13] (BP) models and a procedural learning approach based on simulation and monitoring *learning-by-doing*. To this end, it is of crucial relevance that the procedural aspects of a business process are described in accordance with other relevant views of the social and business context including organizational details, motivational intents, and measure indicators to assess the learning and the process enactment performance.

Model Driven Engineering (MDE) [15] is increasingly gaining acceptance as a mean to leverage business logic and make it resilient to technological changes. Coordinated collections of models and modelling languages are used to describe software systems on different level of abstraction and from different perspectives. In this paper, we describe some the outcomes and the objectives of the Learn PAd WP3: Approaches Enabling Model-based Learning [3]. The workpackage main objective is to define a comprehensive modeling language, called Learn PAd Metamodel (LPMM), for the specification of business processes in Public Administrations consisting of a number of *component metamodels*. Each LPMM metamodel focuses on a different aspect of the business processes. A process is typically perceived as a sequence of activities that the administration executes in order to produce a service for the end user. However, these activities are most of the time knowledge-intensive and require transparency and information tracing. In addition, the responsibility of their enactment is assigned to organizational units within the administrations which pursue given goals. Therefore, in order to better support the learner the typical business process modeling has been intertwined with additional modeling structures to make knowledge relevant in a given process explicit.

Structure of the paper. The paper is organized as follows: in the next section, we describe the main concepts for knowledge-intensive processes in PAs. The model-driven approach is discussed in Sect. 3, it has been used to formalize the Learn PAd metamodels provided in Sect. 4. Finally, in Sec. 5 we show a correspondence between metamodels by means weaving models and in Sect. 6 we draw some conclusions.

2 Processes in Public Administrations

A business process can be regarded as a sequence of activities that the administration executes in order to produce a service for the end user: it starts with i) receiving of some input (i.e. request, documentation), ii) performs activities that add value (i.e., checks) using resources (i.e., humans, information, structures), and finally iii) produce an output. Business processes in PAs are mainly knowledge-intensive. Thus, civil servants are used to dealing with huge amounts of information: lessons learned in previous engagements, insights from prior projects, notes for subsequent process steps are scattered among manifold knowledge containers, from the personal memory, over paper, to different electronic systems. In order to manage such information, it is important to organize knowledge archives exploiting the usage of BPs in a context-giving structure. In particular, the civil servant should be able to access the required knowledge in an optimal manner. This can be achieved by coupling the process model with the descriptive units about various aspects including the kind of data and document type being considered by the process, the organizational structure, the indicators for measuring both the performance of civil servants and how far the learning goals are achieved. The outcome of such modeling procedure is a number of interrelated models as depicted in Fig. 1. The specification of business processes (Fig. 1.a) is usually done by means of standard notations like BPMN 2.0 [13], while for knowledge intensive (sub-)processes (Fig. 1.b) the Case Management Model and Notation (CMMN 1.0) [11] has been considered with some necessary adaptations in order to deal with partiality of information and some intrinsic uncertainty [7]. Another aspect, which is relevant for Learn PAd, is the necessity to consider business goals, Key Performance Indicators (KPIs) and success factors, which are usually represented in a Business Motivation Model (Fig. 1.c) (BMM) [12]. The learning competencies required by roles are modelled as well: the Competency Model (Fig. 1.d), based on the framework CEN [8] is built for describing learners level of competencies, learning progress, etc. Organisational units, roles and persons are modelled in an extra Organisation Model (Fig. 1.e), to allow for more expressiveness than 'BPMN native' pools and lanes, as the BPMN 2.0 specification, for instance, does not provide any semantics to lanes and pools (which are merely regarded as an encapsulation mechanism for organizing activities). Finally a Document Model (Fig. 1.f) is depicted, comprising application documents and data, learning reports etc.

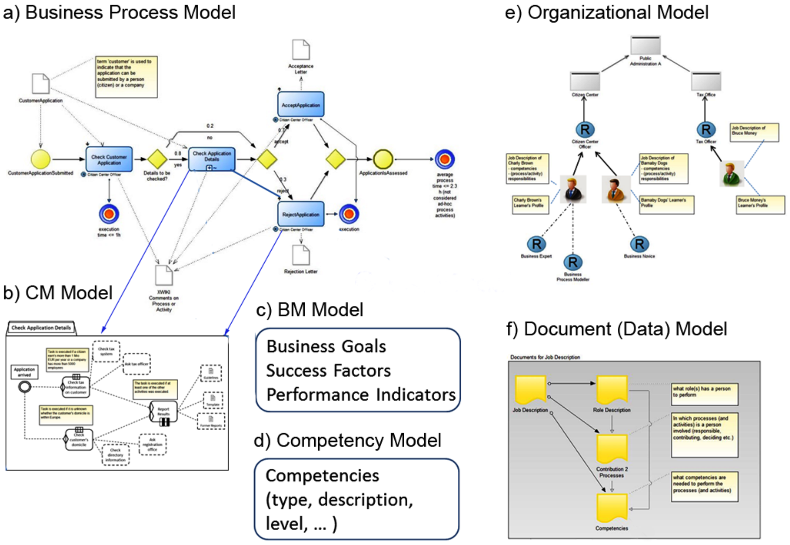

Fig. 1. Overview of needed models

Since, modeling notations are usually given in terms of metamodels a brief introduction to metamodeling and Model-Driven Engineering is given in the next section.

3 Background

In this section we describe the model-driven techniques, which has been used to formalize the Learn PAd metamodels and the correspondences among the different component metamodels.

3.1 Metamodeling

In MDE metamodels consists of a coherent set of interrelated concepts, which are used for formalizing an application domain. A more precise definition, as provided by Seidewitz in [16], is

> *a specification model for a class of system-under-study where each system-under-study in the class is itself a valid model expressed in a certain modeling language.*

Thus, domain instances can be expressed in terms of models, which are said to conform to a metamodel. The expressiveness of the metamodel, i.e., the amount of detail which has to be captured for each concept is a trade-off between abstraction and automation, i.e., the kind of applications (e.g., model-to-model and model-to-code transformations) the designer is aiming at. Being able to define the right generic-specific balance is key to success: a too generic language does not usually offer enough semantic graduation for distinguishing different concepts, on the contrary a too specific language, with too many concepts is difficult to learn, understand, manage, and deploy. In practice, a metamodel often evolves towards a final form only after it undergoes an iterative restructuring and refinement process. Each iteration consists in extending and refining the set of available features and adapting the corresponding model transformations and tools which are tightly coupled with the metamodel.

3.2 Model Weaving

The separation of concerns in software system modeling avoids the construction of large and monolithic models which could be difficult to handle, maintain and reuse. At the same time, having different models (each one describing a certain concern) requires their integration into a final model representing the entire domain [14]. Model weaving can be used in this scenario. Although there is no accepted definition of model weaving, in [4] it is considered as the operation for setting fine-grained relationships between models or metamodels and executing operations on them based on the semantics of the weaving associations specifically defined for the considered application domain.

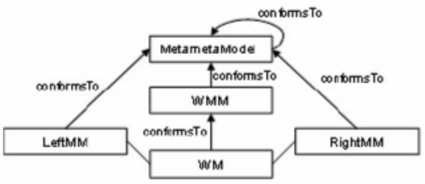

Fig. 2. The analysis process

The definition of model weaving that will be considered in this paper is that provided in [9]. The weaving metamodel is not fixed since it might be extended by means of a proposed composition operation to reach dedicated weaving metamodels. In a weaving model `WM` representing the mapping between the metamodels `LeftMM` and `RightMM` is given. Like other models, this should conform to a specific weaving metamodel `WMM` (see Fig. 2). In the context of Learn PAd we use the weaving models for specifying some form of *semantics* of given modeling elements. For instance, in BPMN the semantics of *lane* is not precisely given, therefore we give a weaving model which can associate a lane to an *organizational unit* deferring the semantics of the former to that of the latter. This technique is a simplification of the *semantic anchoring* [5] which adopts model transformations for anchoring the meaning of a concept in a metamodel into a concept in another metamodel (for which typically the semantics is already given)[6].

4 Learn PAd Platform-Independent Metamodel

The Learn PAd Metamodel (LPMM) is a comprehensive modeling language for the specification of business processes in Public Administrations consisting of a number of component metamodels as illustrated in Fig. 3. The following component metamodels have been defined by adapting current industrial standards:

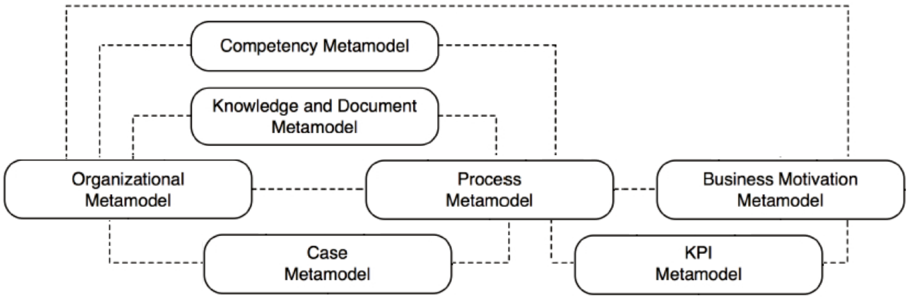

Fig. 3. The Learn PAd metamodel

- business motivation metamodel[7] (BMM) [2],
- business process management and notation (BPMN) [13], and
- case management and notation (CMMN) [1].

Due to space limitations we omit the detailed description of the above metamodels and the interested reader can refer to their standard specifications. The remaining metamodels have been defined from scratch and are described in the following subsections.

- competency metamodel (CM);
- document and knowledge metamodel (DKM);
- key performance indicator metamodel (KPI), and
- organization metamodel (OM).

The dotted lines in the figure denote the correspondences across the different views describing the manifold nature of a process, i.e., concepts belonging to two or more metamodels are cross-linked by means of weaving models [6] (see Sect. 5).

4.1 Competency Metamodel

The Learn PAd Competency Metamodel is thought for describing learners level of competencies and learning progress. In Public Administrations competencies are of-

[6] Weavings are often considered ”declarative transformations” since they define relations from which (specialized) transformations can be automatically derived (see [10]).

[7] In order to stress the distinction between model and metamodel we prefer to use the term metamodel also for denoting standards like OMG’s BMM, which we call Business Motivation Metamodel.

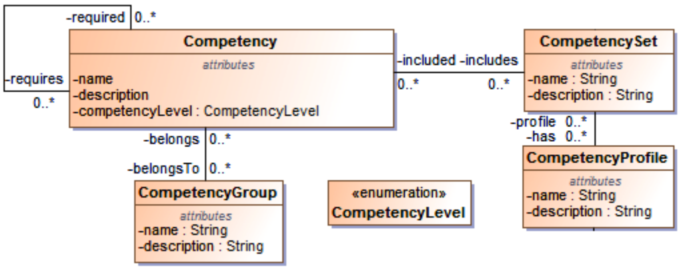

Fig. 4. The Competency metamodel

ten described within job descriptions but not defined in specific models leaving to the modeler the responsibility of consistently meeting them. The Learn PAd Competency Model permits the explicit modeling of such aspect. It is partly based on the framework the European Committee for standardisation, CEN WS-LT LTSO (Learning Technology Standards Observatory)[8]. In particular, it has been simplified in order to avoid any modeling element which is not necessary in the context of Learn PAd. Clearly, the Learn PAd Competency Metamodel plays its crucial role when connected with the business process metamodel desribed above and with the organisation metamodel. Such connections are specified by means of weaving models [6] which represents how model elements in a metamodel correspond to model element in another metamodel by means of typed many-to-many cross-links. An example of the Learn PAd weaving models is given in Sect. 5.

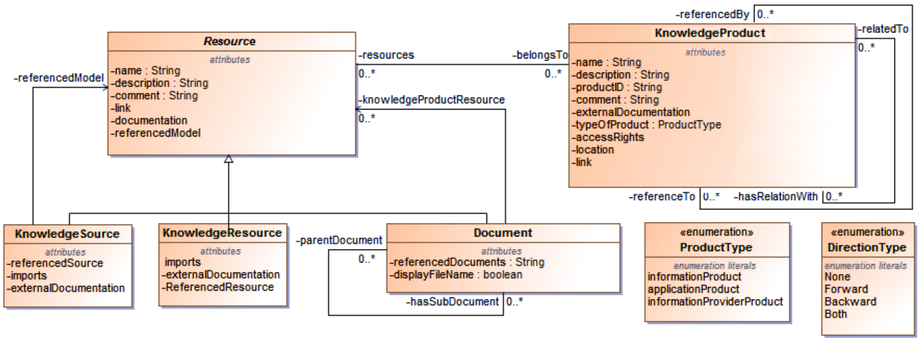

Fig. 5. The Document and Knowledge metamodel

4.2 Document and Knowledge Metamodel

Knowledge models contain documents (templates), knowledge products and resources, which are utilized in the processes (input, output to activities etc.).

Knowledge models can be built hierarchically using document sub models to e.g. illustrate a detailed structure of documents.

[8] EN WS-LT Learning Technology Standards Observatory. URL: http://www.cen-ltso.net/Main.aspx. Main contact: Uni- versity of Vigo 36213 SPAIN.

4.3 The Key Performance Indicator metamodel

Key Performance Indicators (KPIs) are seen as a virtualisation instrument enabling to conceptualise relevant parts of the concrete instances of the production processes. A performance indicator is a measurement of the success of a given organization or activity in which it engages.

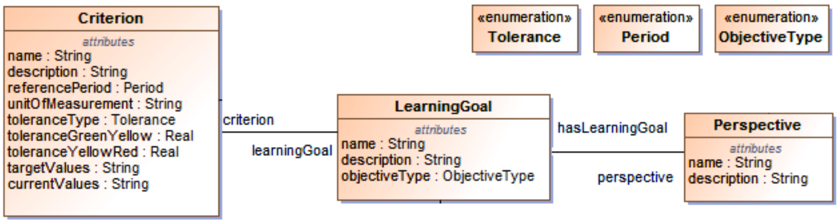

Fig. 6. The KPI metamodel

Thus, KPIs can be successfully employed in process models in order to assess performance of activities and processes. Besides making the KPI modeling explicit by means of the metamodel which is illustrated in this section, we aim at relating KPIs to activities and to the process as a whole. Therefore, proper weaving models will be given in Sect. 5 connecting the KPI metamodel and the business process metamodels and the business motivation metamodel.

In the following the class diagram of the KPI metamodel is given and all the concepts are introduced by specifying the characterizing features.

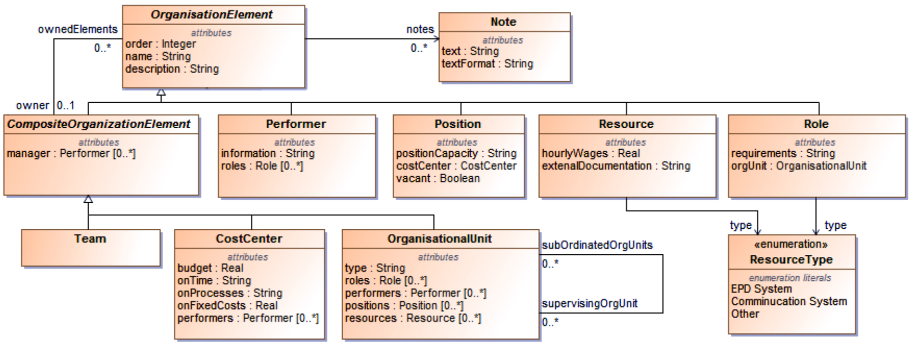

Fig. 7. The Organizational Model

4.4 Organization Model

Organization models describe the structure of an organization (organization chart). Organizational structure models can be built hierarchically using organizational sub models to e.g. illustrate a detailed structure of a working environment.

5 Weaving definition

As already said, weaving models are typically used for defining correspondences between modeling elements belonging to different metamodels and usually depend on each other. In the following, the weaving models, given by means of a weaving metaclass, are used for interconnecting the component metamodels. In the following a fragment of the weaving metamodel which interconnects a BPMN lane in the Perfomer of the OrganizationalUnit metamodel is depicted.

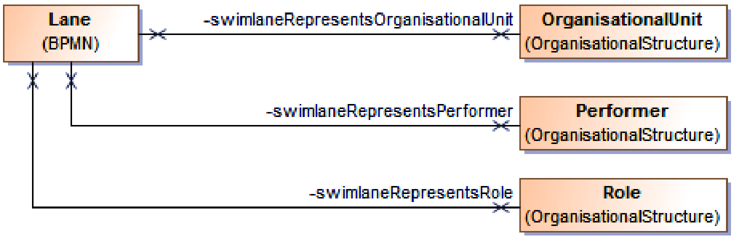
Lane
(BPMN)
-swimlaneRepresentsOrganisationalUnit
OrganisationalUnit
(OrganisationalStructure)
-swimlaneRepresentsPerformer
Performer
(OrganisationalStructure)
-swimlaneRepresentsRole
Role
(OrganisationalStructure)

Fig. 8. The Swimlane-Lane weavings

6 Conclusion

This paper described how business processes in PAs can be described according to the Learn PAd modeling notation: a metamodel stack whose components are model kinds intertwined by means of weaving models. Such specialized models relates the several views a process one with another by means of formal correspondences that can be navigated or used for consistency checks. For instance, it has been possible to specify that the typical concept of a lane in BPMN must correspond to either an organizational unit, a performer, or a role. This is one of the most important criteria we had to follow because BPMN alone does not say anything about the meaning of lane. Future work includes the activities related to the continuation of Task 3.2. This time will be devoted for the design of the modeling tools and for the metamodel validation.

References

1. OMG. Case Management Model and Notation (CMMN), V 1.0. Technical report, Object Management Group OMG, 2013.
2. OMG. Business Motivation Model (BMM). Technical report, Object Management Group OMG, 2014.
3. A. Bertolino. Model-based social learning for public administrations (learn pad). eu-fp7 project. description of work, 2014.
4. J. Bézivin. On the unification power of models. *Software & Systems Modeling*, 4(2):171–188, 2005.
5. K. Chen, J. Sztipanovits, S. Abdelwalhed, and E. Jackson. Semantic anchoring with model transformations. In *Model Driven Architecture–Foundations and Applications*, pages 115–129. Springer, 2005.
6. M. D. Del Fabro, J. Bézivin, F. Jouault, P. Valduriez, et al. Applying generic model management to data mapping. In *BDA*, 2005.

7. R. Eramo, A. Pierantonio, and G. Rosa. Uncertainty in bidirectional transformations. In *Proceedings of the 6th International Workshop on Modeling in Software Engineering*, pages 37–42. ACM, 2014.
8. European Committee For Standardization. A European Model for Learner Competencies. 2005.
9. M.Didonet Del Fabro, J. Bézivin, F. Jouault, and P. Valduriez. Applying Generic Model Management to Data Mapping. In V. Benzaken, editor, *21èmes Journées Bases de Données Avancées, BDA 2005, Saint Malo, Actes (Informal Proceedings)*, 2005.
10. M.Didonet Del Fabro, J.Bezivin, F. Jouault, E. Breton, and G.Gueltas. AMW: A generic Model Weaver. In *Int. Conf. on Software Engineering Research and Practice (SERP05)*, 2005.
11. OMG. Case Management Model and Notation (CMMN), V 1.0. Technical report, Object Management Group OMG, December 2013.
12. OMG. Business Motivation Model (BMM). Technical report, Object Management Group OMG, 2014.
13. B. P. M. OMG. Notation (BPMN) 2.0. *Object Management Group: Needham, MA*, 2494:34, 2011.
14. T. Reiter, E. Kapsammer, W. Retschitzegger, and W. Schwinger. Model Integration Through Mega Operations. 2005. accepted for publication at the Workshop on Model-driven Web Engineering (MDWE2005).
15. D. C. Schmidt. Model-driven engineering. *Computer*, 39(2):25, 2006.
16. E. Seidewitz. What models mean. *IEEE software*, 20(5):26–32, 2003.

OSSMETER: Automated Measurement and Analysis of Open Source Software

Bruno Almeida[1], Sophia Ananiadou[2], Alessandra Bagnato[3], Alberto Berreteaga Barbero[4], Juri Di Rocco[5], Davide Di Ruscio[5], Dimitrios S. Kolovos[6], Ioannis Korkontzelos[2], Scott Hansen[7], Pedro Maló[8], Nicholas Matragkas[6], Richard F. Paige[6], and Jurgen Vinju[8]

[1] UNPARALLEL, Portugal
bruno.almeida@unparallel.pt
[2] National Centre for Text Mining (NaCTeM)
University of Manchester, United Kingdom
sophia.ananiadou,ioannis.korkontzelos@manchester.ac.uk
[3] SOFTEAM, France
alessandra.bagnato@softeam.fr
[4] TECNALIA, Spain
Alberto.Berreteaga@tecnalia.com
[5] Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica
University of L'Aquila, Italy
{juri.diruscio,davide.diruscio}@univaq.it
[6] Department of Computer Science
University of York, United Kingdom
{dimitris.kolovos,nicholas.matragkas,richard.paige}@york.ac.uk
[7] The Open Group, Belgium
s.hansen@opengroup.org
[8] UNINOVA, Portugal
pmm@uninova.pt
[9] Centrum Wiskunde & Informatica, The Netherlands
Jurgen.Vinju@cwi.nl

Abstract. Deciding whether an open source software (OSS) meets the required standards for adoption in terms of quality, maturity, activity of development and user support is not a straightforward process. It involves analysing various sources of information, including the project's source code repositories, communication channels, and bug tracking systems. OSSMETER extends state-of-the-art techniques in the field of automated analysis and measurement of open-source software (OSS), and develops a platform that supports decision makers in the process of discovering, comparing, assessing and monitoring the health, quality, impact and activity of opensource software. To achieve this, OSSMETER computes trustworthy quality indicators by performing advanced analysis and integration of information from diverse sources including the project metadata, source code repositories, communication channels and bug tracking systems of OSS projects.

1 Project data

- **Acronym:** OSSMETER
- **Title:** Automated Measurement and Analysis of Open Source Software

- **Partners:** The Open Group - *Project Coordinator* (Belgium), University of York - *Technical Coordinator* (UK), University of L'Aquila (IT), Centrum Wiskunde & Informatica (NL), University of Manchester (UK), Tecnalia Research and Innovation (ES), UNINOVA (PT), SOFTEAM (FR), Unparallel Innovation (PT)
- **Start date:** 1 October 2012, **Duration:** 30 months
- **Website:** `http://www.ossmeter.eu`

2 Introduction

Deciding whether an open source software (OSS) project meets the required standards for adoption in terms of quality, maturity, activity of development and user support is not a straightforward process; it involves analysing various sources of information – including its source code repositories – to identify how actively the code is developed, which programming languages are used, how well the code is commented, whether there are unit tests etc. Additional information may be pertitent to the analysis, including that from communication channels such as newsgroups, forums and mailing lists to identify whether user questions are answered in a timely and satisfactory manner, to estimate the number of experts and users of the software, its bug tracking system to identify whether the software has many open bugs and at which rate bugs are fixed, and other relevant metadata such as the number of downloads, the license(s) under which it is made available, its release history etc. This task becomes even more challenging when one needs to discover and compare several OSS projects that offer software of similar functionality (e.g., there are more than 20 open source XML parsers for the Java programming language), and make an evidence-based decision on which one should be selected for the task at hand. Moreover, even when a decision has been made for the adoption of a particular OSS product, decision makers need to be able to monitor whether the OSS project continues to be healthy, actively developed and adequately supported throughout the lifecycle of the software development project in which it is used, in order to identify and mitigate any risks emerging from a decline in the quality indicators of the project in a timely manner. Previous work in the field of OSS analysis and measurement has mainly concentrated on analysing the source code behind OSS software to calculate quality indicators and metrics.

OSSMETER extends the scope and effectiveness of OSS analysis and measurement with novel contributions on language-agnostic and language-specific methods for source code analysis, but also proposes using state-of-the-art Natural Language Processing (NLP) and text mining techniques such as question/answer extraction, sentiment analysis and thread clustering to analyse and integrate relevant information extracted from communication channels (newsgroups, forums, mailing lists), and bug tracking systems supporting OSS projects, in order to provide a more comprehensive picture of the quality indicators of OSS projects, and facilitate better evidence-based decision making and monitoring. OSSMETER also provides metamodels for capturing the meta-information relevant to OSS projects, and effective quality indicators, in a rigorous and consistent manner that enable direct comparison between OSS projects. These contributions are integrated in the form of an extensible cloud-based platform through which users can register, discover and compare OSS projects, but which can also be extended in order to support quality analysis and monitoring of proprietary software development projects. To summarize the scientific and technological objectives achieved by OSSMETER are:

– comprehensive *domain modelling* for the domain of open source software development; identification and formal representation of the meta-information that needs to be captured in order to extract meaningful quality indicators for OSS projects;
– extraction of quality metrics by *analysing* aspects related to the *source code* and the development team behind an OSS project;
– extraction of quality metrics related to the *communication channels*, and *bug tracking facilities* of OSS projects using Natural Language Processing and text mining techniques;
– development of an *extensible cloud-based platform* that can monitor and incrementally analyse a large number of OSS projects, and a web-application to present their related quality metrics in an intuitive manner that aids decision making.

In the next sections such objectives are described. For each of them the progress beyond the state of the art is also discussed.

3 Domain Modeling and OSS project Lifecycle Analysis

State of the art: Modeling and abstracting open source software and its management have been the focus of a number of projects and research activities aiming at understanding the current practice in OSS projects e.g., for information and documentation purposes. The Qualipso project[10] analysed many OSS projects in order to identify typical roles (e.g., user, maintainer, and developer), information sources (e.g., help documents, release notes, and source code repositories), and their relations. Qualipso analysed also widely used forges (e.g., SourceForge, and Google Code) in order to identify services, which are typically provided to forge users, and the metadata which is used to describe and support OSS projects. This has been done since there is not a common agreement about the formats and metadata, which have to be used in the whole lifecycle of OSS projects. This hampers the definition of homogeneous treatments of projects maintained in different forges.

Other works (e.g., [9,10]) created abstract models of OSS projects in order to understand their architecture, and their evolution over time. In particular, [9] addresses the structural characteristics of OSS projects, explicitly the organization of the software's constituent components. In [10] the authors, by leveraging the "4+1" view model [17], and the four architectural views of software systems defined in [14], focus on the views which are closer to the work of OSS software developers, such as, for instance, the directory and the file level. The work in [8] proposes models and metrics to support the defect prediction for OSS projects. In particular, in addition to static code attributes for modeling software data in defect prediction, the authors introduce alternative metric sets, such as history and organizational metrics.

To improve both the quality and the trustworthiness perception of OSS products, [20] introduces the idea of certifying the testing process of an OSS system. In this respect,

[10] Qualipso: Leveraging Open Source for Boosting Industry Growth. `http://www.qualipso.org/`

the authors identify peculiar characteristics of OSS projects, that might influence the testing process. The work defines also a certification model that companies, developers, and final users can follow to evaluate the maturity level of an OSS testing process.
Innovation: According to the works previously outlined, the whole life-cycle of OSS projects can be analyzed by means of ad-hoc techniques specifically defined to retrieve heterogeneous information available from different sources in different formats. OSSMETER advances state-of-the-art techniques by providing the means to create models representing in a homogeneous manner different aspects of OSS projects in order to enable objective comparisons of OSS alternatives with respect to user needs, and quality requirements [22]. In particular, OSSMETER has developed:

- Metamodels for the specification of models representing the whole lifecycle of OSS projects. By considering and enhancing existing domain models, a set of EMF/Ecore[11] based metamodels and supporting tools have been conceived in order to enable the representations of OSS projects;
- Metamodels for OSS project metrics to enable automated measurement of open source software.

4 Source Code Quality and Activity Analysis

State of the art: Software metrics are a widely studied subject and are used in practice, for instance in the form of Function Points (FP) to measure the size of software (see International Function Point User Group, IFPUG[12]). Software metrics are widely used for the global analysis of productivity and quality of software [12,15]. All work on activity analysis is ultimately based on the original work of Lehman [18] who also coined the term *software evolution*. There is a wide range of tools available for performing specific analyses on source code as well as for computing various metrics. Regarding analyses, it is not easy to combine the results of different analyses and for metrics the same holds: the results produced by different tools are incomparable since they use different definitions for the underlying metrics. In addition, most of these tools are hand-coded and have to be reimplemented for different languages.
Innovation: OSSMETER provides an integrated view and corresponding tooling to do analyses, metrics calculations and activity analysis on several implementation languages. The main innovation are:

- Definition and of a coherent set of indicators for code quality and activity analysis. These indicators are usable across different implementation languages, different implementation platforms, and different version repository systems;
- Generation of the required tooling from declarative metrics descriptions using innovative model-driven/ generation-based techniques.

5 Communication Channel and Bug Tracking System Analysis

State of the art: Structuring and analysing textual data in forum, newsgroup and community-based question and answer threads is a newly emerging and complex problem

[11] Eclipse EMF: https://www.eclipse.org/modeling/emf/
[12] http://www.ifpug.org/

in text mining. Peer users are the cornerstone of managing software defects in OSS, due to their involvement in online forums [5]. Nevertheless, empirical studies regarding open source quality assurance activities and quality claims are rare [11]. OSS forums and bug-tracking systems concentrate vast amounts of knowledge generated daily about problems and their solutions as well as feedback to requests for OSS improvement.

Mining this textual data can match solutions to problems, evaluate solutions quality and impressively enhance user access to solutions and support [7,21,13]. Due to the size of this textual information, extracting, managing and evaluating it without manual intervention is a demanding, costly, impractical and probably impossible task. Text mining tools that automatically analyse, extract, summarise and assess information found in the threads of discussions on online forums are valuable for supporting OSS. Although text mining techniques have been used extensively in domains such as biomedicine [6], finance [19,16], competitive intelligence [23], very little work has been accomplished on applying text mining techniques for analysing threads of online forums.

Innovation: The target of this analysis is to extract from OSS forums and bug-tracking systems as many indicators about the characteristics and the quality of the communication that takes place as possible. Due to the complexity of the problem, a number of text mining technologies have been combined and structured in levels: after collecting online forum threads, the first level consists of identifying the types of each post as question, answer or supplementary text (context). In succession, posts are classified into more fine-grained categories and similarity-based methods are employed to identify chains of questions, contexts and answers within each thread, i.e. identify which answers and context correspond to which question. Thirdly, posts are analysed as far as sentiment and attitude is concerned. The output of this stage is a fundamental source of evidence useful for quality assessments. Finally, clustering together semantically similar threads and labelling the resulting clusters provides hints about the error-prone aspects of each OSS or its parts that need to be improved. The output of each level is two-fold: a number of indicators about the input posts quality that concerns the specific aspect that the corresponding component exploits; and also, supplementary output useful for the following components, but not necessarily part of the overall system output.

6 OSSMETER platform

State of the art: In the last decade several projects have provided platforms that support automated measurement of open source software including FLOSSMETRICS [1], Qualoss [3], SQO-OSS (Alitheia Core) [4] and Ohloh [2]. Also, many OSS forges such as SourceForge, Google Code and GitHub provide built-in annotation and measurement facilities for the OSS projects they host.

The aim of the FLOSS [13] project was to develop indicators of *non-monetary/trans-monetary* economic activity through a case study of OSS, and to assess OSS business models and best practices, and policy/regulatory impact. Its successor FLOSSMETRICS project [1] integrated a number of source code and bug tracking and mailing list extraction tools into a web-based platform which monitors a selection of open source projects and provides the extracted data in the form of SQL files which then need to be injected into a local database in order to be further analysed.

[13] http://www.flossproject.org/

Qualoss [3] aimed at automating the quality measurement of open source software. The Qualoss platform has been conceived to analyse two types of data: source code and project-repository information and does not appear to be measuring aspects related to communication channels or bug tracking systems of OSS projects.

Alitheia-Core [4] is a platform which aims at enabling software engineering research targeting OSS projects. Alitheia-Core provides support for processing source code repositories, emails from mailing lists and bug tracking systems through an API that developers can use in order to implement *metrics* and *experiments*. The design of Alitheia-Core is similar to the envisioned design of the OSSMETER platform, but the platform itself does not appear to be providing any implemented metrics related to mailing list and bug tracking systems.

Ohloh [2] is a free but proprietary and closed source system, only provided as a hosted service. Ohloh only analyses information related to the source code of OSS projects and does not take communication channels or bug tracking systems into consideration. However, it provides OSS project classification facilities (through user-defined tags), enables OSS project discovery and comparison, and presents source-code and activity-related metrics in an intuitive and understandable manner. On the downside, beyond not taking communication channels and bug tracking systems into consideration, being closed source means that organisations cannot run their own local instance of Ohloh through which they could monitor only the open source projects they are interested in, or their own proprietary projects. Also, as the system is proprietary, developers cannot extend it with features such as support for new metrics, access to additional sources of information, or integration with custom version control management systems.

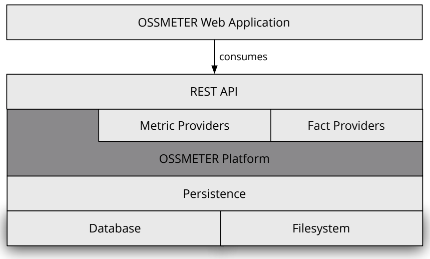

Fig. 1. OSSMETER system architecture

As mentioned above, OSS forges such as SourceForge, Google Code and GitHub provide built-in facilities for capturing additional information (metadata) about projects such as the category they belong to, the languages they are implemented in, relevant news feeds, and activity indicators such as user reviews, number of developers, and number of downloads. However, each OSS forge captures a different set of metadata and as such, projects hosted in different forges are not directly comparable. Moreover, none of these forges provides advanced source code, communication channel, and bug tracking system content analysis features such as those proposed by OSSMETER .

Innovation: The OSSMETER platform integrates and extends components and results produced by the projects discussed above in order to provide the comprehensive system shown in Fig. 1 for analysing and monitoring OSS projects. The novel features of the OSSMETER system are:

▷ a scalable and efficient data storage, which is responsible for storing and retrieving project specific metadata, and metric measurements. The use of local disk storage is

Projects	LOC	Age of Code (days)	# Developers	# Commits	Repository	URL
odoo	2,016,254	2,955	315	93,180	GitHub	`https://github.com/odoo/odoo`
Joomla	865,282	3,465	561	21,831	GitHub	`https://github.com/joomla/joomla-cms`
Drupal	609,987	4,680	107	16,450	GitHub	`https://github.com/drupal/drupal`
Ossmeter	537,343	349	7	1,796	GitHub	`https://github.com/ossmeter/ossmeter`
Assimp	310,235	2,555	77	2,610	GitHub	`https://github.com/assimp/assimp`
Libreplan	291,744	1,251	35	9,346	GitHub	`https://github.com/Igalia/libreplan`
BIMServer	224,000	1,716	15	2,775	GitHub	`https://github.com/opensourceBIM/BIMserver`
Hudson	223,033	3,345	73	1,476	Eclipse	`https://projects.eclipse.org/projects/technology.hudson`
Alitheia-Core	53,874	2,675	11	4,815	GitHub	`https://github.com/istlab/Alitheia-Core`
Epsilon	5,483,784	2,920	6	5,122	Eclipse	`https://projects.eclipse.org/projects/modeling.epsilon`
ATL	563,439	1,684	8	3,661	Eclipse	`https://projects.eclipse.org/projects/modeling.mmt.atl`

Table 1. List of projects used in the evaluation

also enabled to store temporary data required for the analysis, such as clones of source repositories.

▷ support for automated classification of OSS projects and discovery of related projects based on source code, communication channel and bug tracking system analysis through the use of advanced NLP and text mining techniques. To this end different kinds of measure components are provided, namely *fact providers*, *metric providers*, and *factoids*. Fact providers perform utility measurements and store factual data that can be consumed by other fact/metric providers. Metric providers optionally use computed facts to measure one or more project aspects and store the result in the database. Finally, factoids can aggregate heterogeneous metric providers into a four-star system.

▷ an extensible platform implemented using a plug-in based approach (OSGi), which is responsible for the integration of the various OSSMETER components, as well as for their scheduling, execution, and orchestration. The OSSMETER platform is also responsible of mining the OSS data, which are then passed to the various metrics providers for analysis.

▷ a REST API that enables software engineering researchers to access calculated quality indicators in order to perform additional analysis, and developers of 3rd party software to provide added-value services on top of the OSSMETER platform.

▷ a usable web-application developed on top of the platform that enables end-users to explore and compare OSS software in an intuitive manner. The presentation of the information about software projects can be fully customised at the user level and it is based on custom quality models.

7 Conclusion

OSSMETER is officially ended on March 31, 2015. When writing this document, the use case providers were performing the evaluation of the OSSMETER technologies by considering real OSS projects from different application domains. Some of the projects considered during the evaluation are shown in Table 1. These projects were chosen based on their characteristics, such as size, age, number of developers, and number of commits. The code of the OSSMETER platform is publicly available online at `https://github.com/ossmeter/ossmeter`. It is possible to download a locally-deployable version of the OSSMETER system that users can install locally – and if needed extend – in order to monitor a custom selection of OSS projects of interest and/or internal software development projects. By mentioning some facts updated at June 2015, the OSSMETER GitHub repository counted more than 650K lines of code, 1,800 commits, 3 branches, 8 releases, and 8 contributors. More than 30 technical deliverables were produced to present the technologies developed during the project. The official installation of OSSMETER is available at `www.ossmeter.com`.

References

1. FLOSSMETRICS: Free/Libre/Open Source Software Metrics. http://www.flossmetrics.org/.
2. Ohloh Project. http://www.sqo-oss.org/.
3. QUALOSS: Quality in Open Source Software. http://www.qualoss.org/.
4. SQO-OSS: Alitheia Core. http://www.sqo-oss.org/.
5. Faheem Ahmed, Piers Campbell, Ahmad Jaffar, and Luiz Fernando Capretz. Managing support requests in open source software project: The role of online forums. In *ICCSIT 2009*, pages 590–594. IEEE, August 2009.
6. Sophia Ananiadou and John Mcnaught. *Text Mining for Biology And Biomedicine*. Artech House, Inc., Norwood, MA, USA, 2005.
7. Timothy Baldwin, David Martinez, Richard B. Penman, Su N. Kim, Marco Lui, Li Wang, and Andrew MacKinlay. Intelligent linux information access by data mining: the ILIAD project. In *Procs. NAACL HLT 2010*, pages 15–16. Association for Computational Linguistics, 2010.
8. Bora Caglayan, Ayse Bener, and Stefan Koch. Merits of using repository metrics in defect prediction for open source projects. In *Procs. of FLOSS'09*, pages 31–36. IEEE Computer Society, 2009.
9. Andrea Capiluppi and Karl Beecher. Structural complexity and decay in floss systems: An inter-repository study. In *Procs. of CSMR '09*, pages 169–178, 2009.
10. Andrea Capiluppi, Cornelia Boldyreff, and Klaas-Jan Stol. Successful reuse of software components: A report from the open source perspective. In *Open Source Systems: Grounding Research - 7th IFIP WG 2.13 International Conference, OSS 2011*, pages 159–176, 2011.
11. Robert Glass. Is open source software more reliable? an elusive answer. *The Software Practitioner*, 11(6), 2001.
12. R. Grady. *Effective Software Measurements*. Prentice Hall, 1992.
13. Ahmed Hassan, Vahed Qazvinian, and Dragomir Radev. What's with the attitude?: identifying sentences with attitude in online discussions. In *Procs. EMNLP'10*, pages 1245–1255, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
14. Christine Hofmeister, Robert Nord, and Dilip Soni. *Applied Software Architecture*. Addison-Wesley, 2000.
15. Capers Jones. *Applied Software Measurement: Global Analysis of Productivity and Quality, Third Edition*. McGraw Hill, 2008.
16. A. Kloptchenko, T. Eklund, J. Karlsson, B. Back, H. Vanharanta, and A. Visa. Combining data and text mining techniques for analysing financial reports: Research articles. *Int. Journ. of Intelligent Systems in Accounting, Finance and Management*, 12:29–41, January 2004.
17. Philippe Kruchten. The 4+1 view model of architecture. *IEEE Software*, 12(5):88–93, 1995.
18. M.M. Lehman. Programs, life cycles, and laws of software evolution. In *Proceedings IEEE*, volume 68, pages 1060–1976, 1980.
19. Hsin-Min Lu, Hsinchun Chen, Tsai-Jyh Chen, Mao-Wei Hung, and Shu-Hsing Li. Financial text mining: Supporting decision making using web 2.0 content. *IEEE Intelligent Systems*, pages 78–82, 2010.
20. Sandro Morasca, Davide Taibi, and Davide Tosi. Towards certifying the testing process of open-source software: New challenges or old methodologies? In *Procs. FLOSS'09*, pages 25–30, Washington, DC, USA, 2009. IEEE Computer Society.
21. Li Wang, Su N. Kim, and Timothy Baldwin. Thread-level analysis over technical user forum data. In *Procs. of the Australasian Language Technology Association Workshop 2010*, pages 27–31, Melbourne, Australia, December 2010.
22. James R. Williams, Davide Di Ruscio, Juri Di Rocco, and Dimitrios S. Kolovos. Models of OSS Project Meta-Information: A Dataset of Three Forges. In *MSR2014 at ICSE2014*, 2014.
23. Alessandro Zanasi. *Text Mining and its Applications to Intelligence, CRM and Knowledge Management*. WIT Press, 2007.

MONDO: Scalable Modelling and Model Management on the Cloud

Dimitrios S. Kolovos[1], Louis M. Rose[1], Richard F. Paige[1],
Esther Guerra[2], Jesús Sánchez Cuadrado[2], Juan De Lara[2],
István Ráth[3], Dániel Varró[3], Gerson Sunyé[4], Massimo Tisi[4]
{dimitris.kolovos, louis.rose, richard.paige}@york.ac.uk,
{Esther.Guerra, Jesus.Sanchez.Cuadrado, Juan.deLara}@uam.es,
{rath, varro}@mit.bme.hu, {gerson.sunye, massimo.tisi}@inria.fr

[1]University of York,
[2]Universidad Autónoma de Madrid,
[3]Budapest University of Technology and Economics,
[4]AtlanMod (Inria, Mines Nantes, LINA)

Abstract. Achieving scalability in modelling and MDE involves being able to construct large models and domain-specific languages in a systematic manner, enabling teams of modellers to construct and refine large models in collaboration, advancing the state of the art in model querying and transformations tools so that they can cope with large models (of the scale of millions of model elements), and providing an infrastructure for efficient storage, indexing and retrieval of large models. This paper outlines how MONDO, a collaborative EC-funded project, contributes to tackling some of these scalability-related challenges.

1 Project Identity

- **Project acronym:** MONDO
- **Project title:** Scalable Modelling and Model Management on the Cloud
- **Project partners:** The Open Group (Project Coordinator), University of York (Technical Coordinator), Autonomous University of Madrid, University of Nantes, Budapest University of Technology and Economics, IKERLAN, SOFTEAM, Soft-Maint, UNINOVA
- **Website:** `www.mondo-project.org`
- **Project start date/duration:** Nov 1, 2013 (30 months)

2 Introduction

As MDE is increasingly applied to larger and more complex systems, the current generation of modelling technologies are being stressed to their limits in terms of their capacity to accommodate collaborative development, efficient management and persistence of models larger than a few hundreds of megabytes in size. In our view, achieving scalability in MDE involves:

- being able to construct large models and domain specific languages in a systematic manner;
- enabling large teams of modellers to construct and refine large models in a collaborative manner;
- advancing the state of the art in model querying and transformations tools so that they can cope with large models (with millions of model elements);
- providing an infrastructure for efficient storage, indexing and retrieval of such models.

The rest of the paper (Sections 3-6) provides an overview of the state of the art in these four key areas, identifies the main challenges that need to be overcome, and outlines the realised and envisioned contributions of MONDO. Section 7 concludes the paper.

3 Scalable Domain Specific Languages

In current MDE practice, we still find that domain specific modelling languages (DSMLs) are often constructed in an ad-hoc way. Moreover, graphical DSMLs scale poorly for large models. There are several works aimed at defining compositional mechanisms for languages and models. [1] provides composition techniques for languages lacking such built-in mechanisms. Other works extend meta-models [2] and models [3] with export and import interfaces, but are only described theoretically.

With respect to visualizing large models, some researchers have brought techniques from the field of information visualization, like semantic zooming [4]. Some language editors DiaGen enable the definition of abstractions [5], but they have to be manually programmed for each different language. Reusable model abstractions are reported in [6], but with no support for visualizations.

Finally, little work has addressed processes for developing and testing meta-models [7, 8]. In [8] a language to write automated tests for conceptual schemas is proposed, while [7] proposes the incremental development of meta-models by increasingly refined test models. Other works have explored the induction of meta-models from example models [9, 10]. However, none of these works consider issues related to scalability of models or meta-models.

3.1 Research Directions

Scalable Language Design MONDO addresses both the engineering of large DSMLs, and DSMLs expecting large models. For the first issue, we make available reusable patterns, accounting for the several aspects in the definition of a DSML, including abstract, concrete syntax, semantics, and the services of the modelling environment like model fragmentation, filtering and conformance checking. The use of patterns for all these aspects facilitate and speed up DSML definition. We have created a tool, called `DSL-tao` (`http://jdelara.github.io/DSL-tao/`), which permits describing DSMLs and their environments using this philosophy. Second, we propose fragmenting models following modular principles adopted by many programming languages [11]. Therefore a model is organized as a Project, and then be fragmented into Packages (which are mapped to folders in the file system), which may hold Units (mapped in files). This fragmentation strategy is specified at the meta-model level, by instantiating a pattern.

Scalable Visualization Another means to tackle scalability is to support useful abstractions, providing a simplified view of a model, or introducing hierarchical elements, organizing models at different levels of abstraction. In our approach, these abstractions are specified as patterns. Another common issue is that no concrete syntax (only generic tree-based editors) is defined for some meta-models, which becomes problematic as models grow. To address such scenarios, we have created a tool called SAMPLER (http://rioukay.github.io/sampler/), which allows the scalable visual exploration and navigation of EMF models.

Processes and Methodologies Currently, DSMLs are often developed in an informal, ad-hoc way. However, DSMLs should be engineered using sound principles and methods, gathering requirements from all stakeholders. We are currently developing methodologies [12] including validation and verification mechanisms [13, 14] (based on DSLs to specify different kinds of tests) for engineering DSMLs, which are being integrated with the previous tools.

4 Scalable Queries and Transformations

Some work exists that applies either incrementality, laziness or distribution to MDE. *Incremental computation* has been used for transforming large evolving models, either with live [15] (i.e., transformation during the update) or offline [16] (i.e., transformation after the update) incrementality. Incremental graph transformation approaches [17–19] focus especially on techniques for incremental pattern-matching. *Lazy computation* can improve scalability when only a small part of large models is accessed. A model transformation tool with an lazy/on-demand generation strategy has been proposed in [20]. The Stratego [21] system allows user-defined execution strategies for transformation rules, that may be in principle used for on-demand transformation. VIATRA lazily evaluates the matches of connected rules to avoid unnecessary computation [22]. Lazy loading [23] allows to handle models that do not fit into the available memory. *Distributed computation* is convenient for complex and parallelizable transformations. In graph transformation, recent work [24] focuses on parallelizing the recursive matching phase, particularly expensive for graph transformations. In model transformation, Lintra [25] allows to specify distributed transformations by explicit distribution primitives.

4.1 Research Directions

Benchmarks for Scalable Query and Transformation We defined a set of shared benchmarks for query and transformations on very large models, gathering real-world MDE case studies. We have made these cases available to the community for evaluation via a public MDE benchmark repository [26].

Reactive Model Query and Transformation We propose a shift of paradigm for programming MDE applications towards reactive programming [27], where a network of transformations defines persistent data-flows among models. A reactive transformation engine takes care of activating only the strictly needed computation in response to updates or requests of model elements, by a combination of incremental and lazy computation. A reactive engine also opens the way to scenarios based on infinite intermediate models generated on demand, or streaming models propagating from inputs to outputs.

Distributed Model Query and Transformation in the Cloud We propose engines that implicitly (i.e., without explicit distribution primitives) distribute the execution of model queries and transformations on top of well-known distributed programming models for the Cloud (e.g., MapReduce). We show how the execution semantics of model queries and transformations can be aligned with parallel computation models [28].

Prototypes for several components are already available: IncQuery-D, a scalable engine for distributed incremental model queries [29]; MONDO-SAM, a benchmarking framework [30]; VIATRA-CEP, a streaming transformation engine [31].

5 Scalable Collaborative Modelling

Model repositories such as CDO [32] and MORSA [33] are storage systems for modeling artefacts that are mostly focused on concurrent access over a client-server infrastructure. They provide extension mechanisms and core APIs that auxiliary, function-specific tools may use to support conflict management, branching, model comparison etc.

Online collaborative modelling systems such as CoolModes [34] rely on a short transaction approach, whereby a single, shared instance of the model is edited by multiple users concurrently. These systems lack conflict management, or only provide very light weight mechanisms (such as voluntary locking).

Model versioning systems such as EMFStore [35] are more closely aligned with offline version control systems such as SVN. They follow the *long transaction approach* whereby contributors are assumed to commit larger portions of work with respect to a certain (past) version as the reference. Hence, since conflicts are common, their detection, resolution and merging are features of top importance. To that end, such sytems are frequently augmented with *offline model comparison, differencing and merging tools* such as EMF Compare [36] or EMF Diff/Merge [37] are also often used.

The key weaknesses of the collaborative modelling state of the art can be summarized as follows: (i) immature integration of online and offline collaboration patterns; (ii) mostly ad-hoc architectures that prohibit or make the implementation of domain-specific collaboration/version management difficult; (iii) very simplistic locking and conflict management solutions that severely hinder developer productivity; (iv) the lack of a flexible *and* scalable back-end platform that caters to both Eclipse-based and other (commercial) tools.

5.1 Research Directions

We are working on a multi-device collaborative modelling framework which on the front-end is fully compatible with existing and future Eclipse-based technologies (EMF and its auxiliaries and the Team API); on the back-end, and integrates into the scalable model persistence framework. It *supports both offline and online collaboration* in a multi-user and multi-device environment, providing a model access layer (transaction management, queries, views and manipulation) featuring basic collaboration primitives (push, pull, commit, merge), and an adaptation layer for the integration of access control and security services. It is built on an extensible architecture that allows the integration of domain-specific, customized plugins for conflict management (detection, resolution and merging).

As novel and innovative features, it includes *query-driven dynamic locking* that uses complex graph queries [38] for the specification of locking partitions for views and manipulative transactions. Such queries operate in a collaboration-aware manner that includes support for real-time updates and locked queries (where updates are propagated only from a pre-defined subset of collaboration partners). Additionally, the framework features *automated conflict resolution* based on design-space exploration techniques [39] that are able to ensure domain consistency and well-formedness by automatically applying model manipulation policies to find valid and conflict-free model states. Prototypes for both the collaboration framework and the model merger based on design space exploration are already available under `http://github.com/FTSRG/mondo-collab-framework` and `http://github.com/FTSRG/mondo-collab-mergespaceexploration`.

6 Scalable Model Persistence

An essential component of scalable MDE is infrastructure that facilitates persistence and retrieval or large models in an efficient manner.

Efficient Model Storage The current standard model storage format is the XML Metadata Interchange. As XMI is an XML-based format, in order to access any model elements using current state-of-the-art modelling frameworks such as EMF, the complete model file needs to be parsed and loaded in memory first. This implies that the larger the model file, the more time and memory is needed in order to load the model. Also, XMI inherits the verbosity of XML which means that XMI-encoded model files are much larger in size than needed in order to store the information they do.

To address these issues, we envision a new efficient model representation format that will reduce the size of model files, enable modelling and model management tools to lazily load the contents of a model into memory, and access specific model elements without needing to read the entire model file first. We anticipate that such a format will provide a substantial improvement both in terms of both the size of model files, and in terms of the memory and time required to load these models.

Model Indexing With a faster and more efficient model persistence format that provides a reduction of the scale of 10 in terms of size, an XMI-based model of the order of hundreds of MBs, would now be of the order of tens of MBs. In a typical collaborative development environment where artefacts are stored in a central repository (e.g. CVS, SVN, Git etc.), even files of the order of tens of MBs are challenging to manage as for every change they need to be transferred back and forth between the local copy and the remote repository. Storing a large model as a single file can also be sub-optimal as it can cause frequent conflicts when using an optimistic locking VCS or lock-outs when using a pessimistic locking VCS. Two solutions have been proposed for addressing this problem [40]: 1) Storing large models in dedicated model repositories that enable model-element level (instead of file-level) version control operations (check in, check out, lock etc.), and 2) Splitting large models over many cross-referenced physical files (model fragments).

The first approach requires both a leap in terms of the modelling tools used to edit models, as the majority of modelling tools work with file-based models, and a transition from robust and established repositories which work well with a wide range of development tools, to newly developed model-specific repositories. The particularly limited adoption of model-specific repositories such as CDO, and ModelCVS [41] so far has demonstrated that industrial users can be reluctant to make such a drastic transition in practice. As such, and in order to provide industrially-relevant results, we will mainly focus on the second approach. The main advantage of the second approach is that it works well with existing modelling tools, and with existing types of remote repositories (such

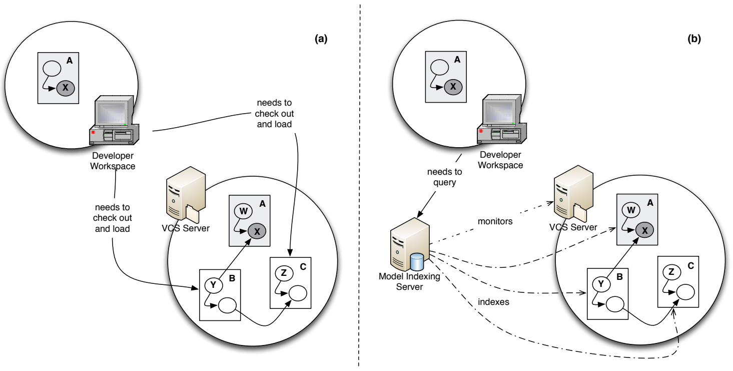

Fig. 1. Performing global queries on model fragments stored in a VCS repository without (a) and with (b) an indexing server

as CVS, SVN, Git, FTP, shared network folders etc). However, using this approach with current state-of-the-art technologies makes it impossible to compute queries of global nature without going through all the model fragments from the remote repository every time. For example, consider the scenario on the left side (a) of Figure 1, where the VCS repository contains 3 model fragments (A, B and C) from which the developer has checked out only fragment A. Now, if the developer needs to know which other fragments in the repository reference its X element, they need to check out, load and examine every other fragment in the repository (B and C in this case). Obviously, as the number of model fragments in the repository grows, this approach becomes increasingly inefficient.

To address this limitation, we are working on a model indexing framework (Hawk [42] - `https://github.com/kb634/mondo-hawk`) that can monitor the contents of remote version control repositories, and index the models they contain in a scalable database that will enable efficient computation of global queries. Hawk can support monitoring different types of remote repositories (SVN, Git etc.) and indexing of heterogeneous models (i.e. XMI, proprietary) using a driver-based architecture.

7 Conclusions and Next Steps

In this paper we have provided an outline of the main scalability challenges in MDE, and MONDO's technical vision for addressing them. MONDO has already contributed novel techniques and several prototype implementations in all four identified key-areas. During the last year of the project, we plan to increasingly focus on integrating these prototypes in the context of a unified technical offering in preparation for the evaluation phase of the project, where the research contributions of MONDO will be assessed in the context of four industrial case studies.

The first case study (provided by UNINOVA[1]) comes from the construction industry and involves collaborative development and automated management of large computer-aided design (CAD) models. The second case study (provided by Soft-Maint[2]) involves exploration and automated transformation of large models which have been reverse-engineered from existing codebases. The third case study (provided by IKERLAN[3]) involves multi-device collaborative development of models from the offshore wind power industry, and the fourth case study involves managing large collections of UML models stored in a proprietary format supported by Softeam's[4] Modelio[5] tool.

[1] `http://www.uninova.pt/`

[2] `http://www.sodifrance.fr/`

[3] `http://www.ikerlan.es/`

[4] `http://www.softeam.fr/`

[5] `https://www.modeliosoft.com/`

References

1. Florian Heidenreich, Jakob Henriksson, Jendrik Johannes, and Steffen Zschaler. On language-independent model modularisation. *T. Aspect-Oriented Software Development VI*, 6:39–82, 2009.
2. Ingo Weisemöller and Andy Schürr. Formal definition of MOF 2.0 metamodel components and composition. In *Model Driven Engineering Languages and Systems, 11th International Conference*, volume 5301 of *Lecture Notes in Computer Science*, pages 386–400. Springer, 2008.
3. Stefan Jurack and Gabriele Taentzer. A component concept for typed graphs with inheritance and containment structures. In *Graph Transformations - 5th International Conference, ICGT 2010*, volume 6372 of *Lecture Notes in Computer Science*, pages 187–202. Springer, 2010.
4. Mathias Frisch, Raimund Dachselt, and Tobias Brückmann. Towards seamless semantic zooming techniques for UML diagrams. In *SOFTVIS*, pages 207–208. ACM, 2008.
5. Oliver Köth and Mark Minas. Structure, abstraction, and direct manipulation in diagram editors. In *Diagrams*, volume 2317 of *LNCS*, pages 290–304. Springer, 2002.
6. Juan de Lara, Esther Guerra, and Jesús Sánchez Cuadrado. Abstracting modelling languages: A reutilization approach. In *Proc. CAiSE 2012*, volume 7328 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2012.
7. Antonio Cicchetti, Davide Di Ruscio, Dimitris Kolovos, and Alfonso Pierantonio. A test-driven approach for metamodel development. In *Emerging Technologies for the Evolution and Maintenance of Software Models*, pages 319–342. IGI Global, 2012.
8. Albert Tort and Antoni Olivé. An approach to testing conceptual schemas. *Data Knowl. Eng.*, 69(6):598–618, 2010.
9. Hyun Cho, Yu Sun, Jeff Gray, and Jules White. Key challenges for modeling language creation by demonstration. In *ICSE'11 Workshop on Flexible Modeling Tools*, 2011.
10. Jesús Sánchez Cuadrado, Juan de Lara, and Esther Guerra. Bottom-up metamodelling: An interactive approach. In *Model Driven Engineering Languages and Systems - 15th International Conference, MODELS 2012, Innsbruck, Austria, September 30-October 5, 2012. Proceedings*, volume 7590 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2012.
11. Antonio Garmendia, Esther Guerra, Dimitrios S. Kolovos, and Juan de Lara. EMF splitter: A structured approach to EMF modularity. In *XM@MoDELS*, volume 1239 of *CEUR Workshop Proceedings*, pages 22–31. CEUR-WS.org, 2014.
12. Jesús Juan López-Fernández, Jesús Sánchez Cuadrado, Esther Guerra, and Juan de Lara. Example-driven meta-model development. *SoSyM*, in press, 2014, see also `http://www.miso.es/tools/metaBUP.html`.
13. Jesús J. López-Fernández, Esther Guerra, and Juan de Lara. Assessing the quality of meta-models. In *MoDeVVa@MODELS*, volume 1235 of *CEUR Workshop Proceedings*, pages 3–12. CEUR-WS.org, 2014.
14. Jesús J. López-Fernández, Esther Guerra, and Juan de Lara. Meta-model validation and verification with metabest. In *ACM/IEEE ASE*, pages 831–834. ACM, 2014.
15. Frédéric Jouault and Massimo Tisi. Towards incremental execution of ATL transformations. In *Theory and Practice of Model Transformations*, pages 123–137. Springer, 2010.

16. Yingfei Xiong, Dongxi Liu, Zhenjiang Hu, Haiyan Zhao, Masato Takeichi, and Hong Mei. Towards automatic model synchronization from model transformations. *Proc. of ASE'07*, page 164, 2007.
17. G. Bergmann, I. Ráth, and D. Varró. Parallelization of graph transformation based on incremental pattern matching. *Electronic Communications of EASST*, 18, 2009.
18. Gabor Bergmann, Istvan Rath, Gergely Varro, and Daniel Varro. Change-driven model transformations. *Software and Systems Modeling*, pages 1–31, 2011. 10.1007/s10270-011-0197-9.
19. Holger Giese and Robert Wagner. From model transformation to incremental bidirectional model synchronization. *Software & Systems Modeling*, 8(1):21–43, 2008.
20. Massimo Tisi, Salvador Martínez, Frédéric Jouault, and Jordi Cabot. Lazy execution of model-to-model transformations. In *Model Driven Engineering Languages and Systems*, pages 32–46. Springer, 2011.
21. Eelco Visser. Program transformation with Stratego/XT: Rules, strategies, tools, and systems in Stratego/XT 0.9. In *Domain-Specific Program Generation*, volume 3016 of *LNCS*, pages 216–238. Springer, 2003.
22. Gabriele Taentzer, Karsten Ehrig, Esther Guerra, J. de Lara, L. Lengyel, Tihamer Levendovszky, Ulrike Prange, D. Varró, and S. Varró-Gyapay. Model transformation by graph transformation: A comparative study. In *Proc. Workshop Model Transformation in Practice*, 2005.
23. Jouault, Frédéric and Bézivin, Jean and Barbero, Mikaël. Towards an advanced model-driven engineering toolbox. *Innovations in Systems and Software Engineering*, 5:5–12, 2009.
24. Gergely Mezei, Tihamer Levendovszky, Tamas Meszaros, and Istvan Madari. Towards Truly Parallel Model Transformations: A Distributed Pattern Matching Approach. In *IEEE EUROCON 2009*, pages 403–410. IEEE, 2009.
25. Loli Burgueño, Javier Troya, Manuel Wimmer, and Antonio Vallecillo. On the Concurrent Execution of Model Transformations with Linda. In *Proceeding of the First Workshop on Scalability in MDE*, BigMDE '13, pages 3:1–3:10, New York, NY, USA, 2013. ACM.
26. Amine Benelallam, Massimo Tisi, Istvan Rath, Benedeck Izso, and Dimitrios S. Kolovos. Towards an Open Set of Real-World Benchmarks for Model Queries and Transformations. In CEUR Workshop Proceedings, editor, *BigMDE*, York, UK, United Kingdom, July 2014. University of York.
27. D. Harel and A. Pnueli. *On the development of reactive systems*, pages 477–498. Springer-Verlag New York, Inc., New York, NY, USA, 1985.
28. Massimo Tisi, Frdric Jouault, Jrme Delatour, Zied Saidi, and Hassene Choura. fuml as an assembly language for model transformation. In Benot Combemale, DavidJ. Pearce, Olivier Barais, and JurgenJ. Vinju, editors, *Software Language Engineering*, volume 8706 of *Lecture Notes in Computer Science*, pages 171–190. Springer International Publishing, 2014.
29. Gábor Szárnyas, Benedek Izsó, István Ráth, Dénes Harmath, Gábor Bergmann, and Dániel Varró. Incquery-d: A distributed incremental model query framework in the cloud. In *ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems, MODELS 2014*, Valencia, Spain, 2014. Springer, Springer. Acceptance rate: 26%.
30. Benedek Izsó, Gábor Szárnyas, István Ráth, and Dániel Varró. Mondo-sam: A framework to systematically assess mde scalability. In *BigMDE 2014 2nd Workshop on Scalable Model Driven Engineering*, page 40. ACM, ACM, 2014.

31. István Dávid, István Ráth, and Dániel Varró. Streaming model transformations by complex event processing. In Juergen Dingel and Wolfram Schulte, editors, *ACM/IEEE 17th International Conference on Model Driven Engineering Languages and Systems, MODELS 2014*, Valencia, Spain, 2014. Springer, Springer. Acceptance rate: 26%.
32. Eclipse. The connected data objects model repository (CDO) project, 2012. `http://eclipse.org/cdo`.
33. Javier Espinazo Pagan, Jesus Sanchez Cuadrado, and Jesus García Molina. Morsa: A scalable approach for persisting and accessing large models. In Jon Whittle, Tony Clark, and Thomas Kühne, editors, *Model Driven Engineering Languages and Systems*, volume 6981 of *Lecture Notes in Computer Science*, pages 77–92. Springer Berlin / Heidelberg, 2011. 10.1007/978-3-642-24485-8_7.
34. Niels Pinkwart. A Plug-In Architecture for Graph Based Collaborative Modeling Systems. In V. Aleven et al, editor, *Supplementary Proceedings of the 11th Conference on Artificial Intelligence in Education, Sydney (Australia)*, pages 89–94, Sydney, Australia, 2003. SIT.
35. Maximilian Koegel and Jonas Helming. Emfstore: a model repository for emf models. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering - Volume 2*, ICSE '10, pages 307–308, New York, NY, USA, 2010. ACM.
36. C. Brun et al. EMF compare 2.0, 2012. `http://www.eclipse.org/emf/compare/`.
37. O. Constant. EMF Diff/Merge, 2012. `http://eclipse.org/diffmerge/`.
38. Zoltán Ujhelyi, Gábor Bergmann, Ábel Hegedüs, Ákos Horváth, Benedek Izsó, István Ráth, Zoltán Szatmári, and Dániel Varró. Emf-incquery: An integrated development environment for live model queries. *Science of Computer Programming*, 98, 02/2015 2015.
39. Ábel Hegedüs, Ákos Horváth, and Dániel Varró. A model-driven framework for guided design space exploration. *Automated Software Engineering*, pages 1–38, 08/2014 2014.
40. Dimitris Kolovos and Richard Paige and Fiona Polack. The Grand Challenge of Scalability for Model Driven Engineering. In Chaudron, Michel, editor, *Models in Software Engineering*, volume 5421 of *Lecture Notes in Computer Science*, pages 48–53. Springer Berlin / Heidelberg, 2009.
41. G. Kramler, G. Kappel, T. Reiter, E. Kapsammer, W. Retschitzegger, and W. Schwinger. Towards a semantic infrastructure supporting model-based tool integration. In *Proceedings of the 2006 international workshop on Global integrated model management*, GaMMa '06, pages 43–46, New York, NY, USA, 2006. ACM.
42. Konstantinos Barmpis and Dimitrios S. Kolovos. Towards scalable querying of large-scale models. In Jordi Cabot and Julia Rubin, editors, *Modelling Foundations and Applications*, volume 8569 of *Lecture Notes in Computer Science*, pages 35–50. Springer International Publishing, 2014.

SeaClouds: Agile management of complex applications across multiple heterogeneous clouds

D. Athanasopoulos[1], M. Barrientos[2], L. Bartoloni[3], A. Brogi[3],
M. Buccarella[3], J. Carrasco[2], J. Cubo[2], F. D'Andria[4],
E. Di Nitto[1], A. Nieto[2], M. Oriol[3], E. Pimentel[2], S. Zenzaro[3]

[1] Dipartimento di Elettronica, Informazione e Bioingegneria,
Politecnico di Milano, Italy
[2] Departamento de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, Spain
[3] Department of Computer Science, University of Pisa, Italy
[4] ATOS, Spain

Abstract. SeaClouds is a European FP7 research project, whose goal is to develop a novel open solution to provide developers with the capabilities to design, configure, deploy, and manage complex applications across multiple heterogeneous clouds in an efficient and adaptive way.

1 General data of the project

Name	Seamless adaptive multi-cloud management of service-based applications
Acronym	SeaClouds
Project Id	EU-FP7-ICT-610531
Funding source	European Union
Duration	From October 1, 2013 to March 31, 2016
Total funding	Budget: 2.99 MEuros - EU contribution: 2.19 MEuros
Consortium	ATOS (Spain)**,** University of Malaga (Spain), University of Pisa (Italy), Politecnico di Milano (Italy), Cloudsoft (UK), Nurogames (Germany)
Web site	http://www.seaclouds-project.eu/

2 Brief description of the project

How to deploy and manage, in an efficient and adaptive way, complex applications across multiple heterogeneous cloud platforms is one of the problems that have emerged with the cloud revolution. SeaClouds is a European FP7 research project which aims at enabling a seamless adaptive multi-cloud management of complex applications by supporting the distribution, monitoring and migration of application modules over multiple heterogeneous cloud platforms. In short:

- SeaClouds is a software platform which supports the “design, development, planning and management” of complex business applications distributed on multi-cloud environments.

- SeaClouds provides an enterprise capability with continuous software delivery that enables independent application vendors to mitigate risks and reduce time and cost to market.
- SeaClouds orchestrates services, platforms and infrastructures to ensure that they dynamically meet the needs of cloud applications.
- SeaClouds provides an integrated standards-based multi-cloud application management system that follows the DevOps approach with various basic capabilities delivered to the developer via an innovative Graphical User Interface (GUI).

3 Objectives of the project

The SeaClouds project aims to develop a new open source platform featuring a seamless adaptive multi-cloud management of service-based applications. The platform consists of an application management system over IaaS and PaaS clouds, and it implements a DevOps approach for continuous software delivery.

SeaClouds permits developers to design, deploy, manage and configure complex applications across multiple and heterogeneous clouds, something unfeasible hitherto.

The specific objectives of SeaClouds are:

- ***Orchestration and adaptation of services distributed over different cloud providers***. SeaClouds aims at providing the assisted design, synthesis, and simulation of service orchestrations on different cloud providers, by distributing modules of cloud-based applications over multiple heterogeneous clouds.
- ***Unified application management of services distributed over different clouds***. SeaClouds will be able to deploy, manage, scale and monitor services over technologically diverse cloud providers. Such operations will be performed by taking into account application requirements and by providing developers with support beyond the handling of single services.
- ***Monitoring and run-time reconfiguration operations of services distributed over multiple heterogeneous cloud providers***. Monitoring will be in charge of detecting the possible need of redistributing services across cloud providers. Dynamic reconfiguration will let orchestrations evolve so as to realize all the required changes. Reconfiguration ranges from dynamically replacing malfunctioning services to migrating them to different cloud providers.
- ***Compliance with major standards for cloud interoperability***. SeaClouds will manage applications deployed on technologically diverse cloud platforms, unifying operations such as monitoring and lifecycle management, promoting the adoption of OASIS standards for cloud interoperability, in particular TOSCA[1] and CAMP[2].

[1] https://www.oasis-open.org/committees/tosca/

[2] https://www.oasis-open.org/committees/camp/

4 Final expected outcomes of the project

In order to achieve the aforementioned objectives, the following capabilities are expected as outcomes of SeaClouds:

- ***Matchmaking***. To allow querying or browsing available (IaaS and PaaS) cloud offerings and select them based on application requirements.
- ***Deployment optimizer***. To optimize the deployment topology of an application across multiple clouds to address non-functional requirements.
- ***Application management***. To support efficient deployment and multi-cloud governance of complex applications on various (IaaS and PaaS) cloud offerings by leveraging cloud harmonized APIs and platform-specific adapters.
- ***Monitoring and SLA enforcement***. To feature monitoring and independent metrics to allow users to monitor the health and performance of applications.
- ***Repairing***. To scale both horizontally and vertically cloud resources to maximize the performance of each module of an application.
- ***Replanning and application migration***. To provide a seamless migration of different application modules between dissimilar clouds.
- ***Database migration and data synchronization***. To enable the portability of data among databases on different clouds.

5 Current status of the project

We now describe the platform architecture, SeaClouds main functionalities, the case studies used to validate the platform, and relations and collaborations with other projects, initiatives and standards.

5.1 Platform architecture

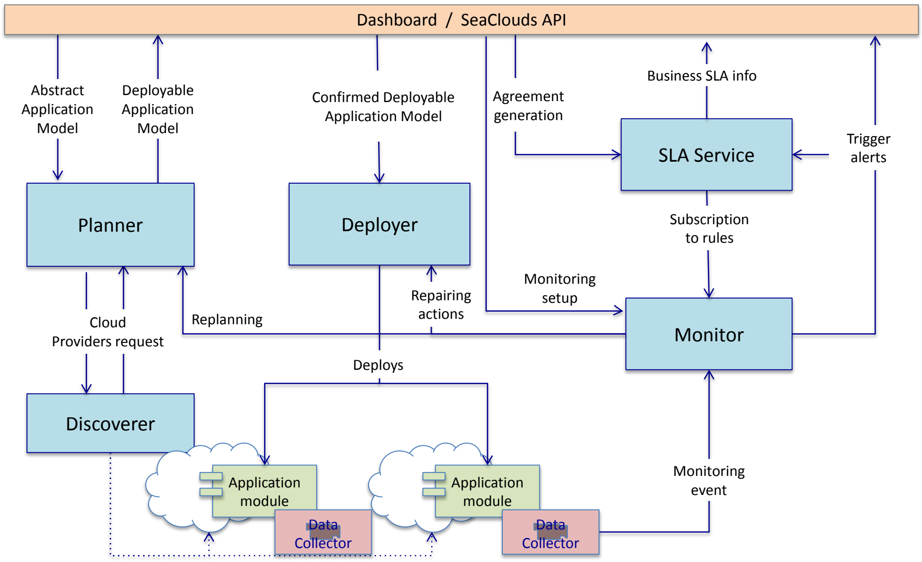

Figure 1. SeaClouds architecture.

Figure 1 illustrates the architecture and main components of SeaClouds. On top of the figure, there is the ***Dashboard / SeaClouds API***, which provides a harmonized API and a dashboard to manage cloud applications with SeaClouds. This component orchestrates the different modules of the SeaClouds platform:

- ***SeaClouds Discoverer***. It is in charge of discovering available capabilities and add-ons offered by available cloud providers.
- ***SeaClouds Planner***. It is in charge of generating an orchestration plan taking into account the application topology and requirements.
- ***SeaClouds Deployer***. It is in charge of executing deployment plans generated by the Planner.
- ***SeaClouds Monitor***. It is in charge of monitoring that the Quality of Services (QoS) properties of the application are not violated by the clouds in which the application modules were deployed, and of determining -together with the Deployer- the reconfiguration strategies to trigger repairing or replanning actions.
- ***SeaClouds SLA Service***. It is in charge of mapping low level information gathered from the Monitor into Quality of Business (QoB) information about the fulfilment of the SLA defined.

A distinguishing aspect of the SeaClouds architecture is that it builds on top of two OASIS standards initiatives: TOSCA, to specify the topologies and deployment plans, and CAMP, to interoperate with heterogeneous PaaS providers.

5.2 Design-time functionalities

The supported design time functionalities are:

- ***Matchmaking***, implemented by the Planner component,
- ***Deployment optimizer***, implemented by the Planner component,
- ***Discovery*,** implemented by the Discoverer component.

At design time, the designer specifies the Abstract Application Model (AAM), which is a TOSCA YAML model that defines the topology and requirements of the application to be deployed. Once the AAM has been specified, SeaClouds starts the Planning stage. The Planner acts with two sub-processes: Matchmaking and Optimizer (as described in D3.1[3]).

The matchmaking aims at finding, for each deployable module in the AAM, the list of candidate cloud offerings that fulfil the requirements for that module. To generate such list of cloud offerings, the planner interacts with the Discoverer. Then, the Optimizer selects one cloud offering from the list of candidates using metaheuristic algorithms. As a result, the Planner generates a Deployable Application Model (DAM), which specifies the concrete cloud services used to distribute the application.

[3] SeaClouds Deliverable D3.1 - Discovery, design and orchestration functionalities: First specification and prototype. Available from http://www.seaclouds-project.eu/deliverables.html

5.3 Run-time functionalities

The supported run-time functionalities are:

- ***Application management***, implemented by the Deployer component.
- ***Monitoring and SLA enforcement***, jointly implemented by the Deployer, Monitoring and SLA Service components.
- ***Repairing***, jointly implemented by the Deployer and Monitor components.
- ***Replanning*** and ***application migration***, jointly implemented by the Deployer, Monitor, and Planner components.
- ***Database migration and data synchronization***, integrated from the MODAClouds EU project[4] and is integrated with the other elements of SeaClouds so that they can activate it.

The Deployer component, detailed in Deliverable D4.1[5], executes the confirmed DAM, while the Monitor is configured with the monitoring rules taken from the user requirements. Also, the SLA service subscribes to rules or alerts and in connection with the Monitor, it enforces the policies of the agreements. The Deployer permits the deployment of the application's modules over heterogeneous IaaS and PaaS, and it tracks the dynamic evolution of the deployment and management of the application modules themselves, by means of a Live Application Model (LAM), as described in Deliverable D4.1.

Once the application is deployed, the Deployer manages it and instruments the Monitor. In particular, the Deployer installs DataCollectors in the cloud machine(s) in which the application modules have been deployed. A DataCollector component gathers raw monitoring data and pushes them to the Monitor. The latter component interacts with the SLA service to manage possible violations of QoS and QoB properties.

When a violation issue occurs, and it can be fixed without replanning, the Monitor and the Deployer interact to repair the issue. Otherwise, the Monitor interacts with the Planner to trigger the generation of a new plan to be executed by the Deployer.

5.4 Case studies

5.4.1 ATOS case study

The ATOS case study is a complex software application that aims at developing an innovative and integrated solution for the use of social inclusion tools by elderly people and for the general management (self-management included) of their medical problems. The currently existing application will be decomposed into a set of modules that will be deployed, monitored and administered in one or more cloud (IaaS and PaaS) cloud providers with the support featured by the SeaClouds platform.

From the re-engineering our health application, ready to be ported on the cloud through the SeaClouds platform, we expect three main benefits:

[4] http://www.modaclouds.eu

[5] SeaClouds Deliverable D4.1 - Definition of the multi-deployment and monitoring strategies. Available from http://www.seaclouds-project.eu/deliverables.html

- To reduce the vendor lock-in problem, by having the opportunity to choose and change competitive providers according to the changing resource requirements during the lifetime of the application,
- To execute the application on fully managed cloud infrastructure running in enterprise-class data centers with secure, reliable cloud services to support and deliver hosted software solutions, and
- To deliver flexible resources to accommodate fluctuations in customers' production workloads as well as on-demand capacity for training, demonstration, proof-of-concept or test/development.

5.4.2 NURO case study

NURO cloud game case study is based on a Nurogames Engine, which is responsible for data consistency and cheating protection. NURO will modify its monolithic server approach to get a cloud ready version.

The goal of the case study is to keep the functionalities of the currently available version, by taking advantage of cloud flexibility and of SeaClouds tools to optimize and to save administrative time and effort. Reliability, stability and security of the game server application should be improved by the SeaClouds solution. Saving resources in burst scenarios and a flexible pricing of real used resources will be possible.

The main benefits that we expect from porting NURO application to the cloud via SeaClouds are:

- Stable QoS. The flexible use of resources should guarantee the QoS on pick phases. Maintenance time should be optimized on rollout of new versions.
- Simpler administration and maintenance. Designers and administrators can focus on the application. They just have to configure the modules according to their needs and invoke deployment, installation and updates of the application. Administration of hardware and operating system is to be done by the SeaClouds system or the cloud provider.
- Scalability over time. If the game is accepted and running for a long time, increasing number of players will cause the server applications to consume more resources. A cloud solution should make it easy to scale the system to the users' needs. Moreover, on pick phases the QoS limits will be met and on burst phases wasting of resources will be avoided.
- Flexibility and control on payment. SeaClouds could considerably reduce costs, only paying for resources which are actually needed. Consuming and releasing resources depends on the needs to satisfy the QoS rules. We expect to set also cost limits to avoid exceeding budgets.

5.5 Relations and collaborations with other projects and standards

As discussed earlier, SeaClouds aims at covering different needs of application deployment and management over multiple IaaS and PaaS cloud providers.

As far as IaaS environments are concerned, the SeaClouds platform takes into account the main IaaS APIs and technologies, such as OpenStack[6] and CloudStack[7]. To solve the problem of deploying multi-cloud in IaaS environments, SeaClouds is using Apache jclouds[8] via Apache Brooklyn[9]. Recently, SeaClouds members have been contributing on adding the support for Docker and Microsoft Azure[10] compute to Apache jclouds.

As far as PaaS environments are concerned, cloud developers are often locked into a specific cloud environment. To mitigate the vendor lock-in issues, several research projects try to wrap services offered by different providers in order to make them compatible. One of the most related projects in this regard is Cloud4SOA[11], which allows the management of PaaS services of different providers. Also, Cloud4SOA provides a set of cloud-independent monitoring mechanisms and a matchmaking methodology. SeaClouds takes advantage of Cloud4SOA and moves beyond it by defining its own matchmaking mechanisms for selecting target services that better adapt to the application components.

Another related project is CloudWave[12]. With CloudWave, service providers will be able to rapidly design and deliver innovative, sustainable digital services for consumers at low cost and high quality. SeaClouds converges with CloudWave in the used distributed algorithms and data models that enable cloud infrastructures and applications to take actions in response to the dynamic changes in their environment. On the other hand, SeaClouds focuses on tackling the problem of deploying and managing complex multi-services applications over technologically heterogeneous clouds environments. It is worth mentioning SeaClouds and CloudWave organize a joint scientific workshop at ESOCC 2015[13].

Also, SeaClouds relates to MODAClouds[14], which aims at providing quality assurance during the application's life-cycle, mitigating part of the portability vendor lock-in issues. To achieve its purposes, MODAClouds offers a monitoring methodology independent of any provider API. This monitoring methodology is going to be extended by SeaClouds project.

The peculiarity of SeaClouds is in the development of an almost market-ready product, adopting and extending Brooklyn (whose main developers are part of the

[6] https://www.openstack.org

[7] https://cloudstack.apache.org

[8] https://jclouds.apache.org

[9] https://brooklyn.incubator.apache.org

[10] http://azure.microsoft.com

[11] http://www.cloud4soa.com

[12] http://cloudwave-fp7.eu

[13] http://esocc2015.unime.it

[14] http://www.modaclouds.eu

project) and employing OASIS TOSCA[15] and CAMP[16], as main standards for the deployment and management of cloud applications. SeaClouds members have been involved with CAMP and TOSCA technical committees, suggesting improvements to the specifications based on our activity. Also, SeaClouds members attend conference calls being held within the TOSCA monitoring subgroup, which focuses on extending TOSCA specification to describe monitoring aspects of an application.

Last, but not least, a cross-fertilization effort between Alien4Cloud[17] and SeaClouds is currently ongoing to assess the possibility for Alien4Cloud to use SeaClouds deployer, and for SeaClouds to use Alien4Cloud TOSCA YAML parser.

A more thorough discussion of related projects is provided in deliverables D2.3.1[18] and D2.3.2[19].

6 Impact and Exploitation

SeaClouds adopted from the very beginning of the project an **open source strategy**, which includes transparent development, Apache 2.0 licensing, and developer engagement. This strategy also serves as enabler for three main impact paths:

- **Apache Brooklyn Integration and Extension for Application Deployment.** The Apache Brooklyn application management framework will not only serve as part of SeaClouds' Deployer component, but also as a vehicle for SeaClouds' post-project value proposition by supporting TOSCA and extending deployment capabilities to PaaS layer and multi-cloud scenarios. The SeaClouds-extended Apache Brooklyn will offer a more robust multi-cloud application deployment for European service providers to accompany their IaaS and PaaS service, strengthening a solution already supported in the market.
- **Contribution to Vendor-Driven Standards.** SeaClouds is actively contributing to two leading standards in cloud computing PaaS segment: OASIS CAMP and TOSCA. By extending these standards, SeaClouds is tapping a channel directly to Europe's cloud supply (vendors) to help empower multi-cloud scenarios for demand (their customers) via improved topology, orchestration, management and interoperability.
- **Consortium Exploitation Opportunities.** Project partners will be able to exploit SeaClouds' open-source solution to strengthen their commercial and research portfolios. Industry partners such as Cloudsoft and ATOS can integrate SeaClouds assets into their own selected commercial offerings, benefiting as primary contributors to development. Similarly, academic partners can propel their research in the multi-cloud domain.

[15] https://www.oasis-open.org/committees/tosca

[16] https://www.oasis-open.org/committees/camp

[17] http://alien4cloud.github.io

[18] SeaClouds Deliverable D2.3.1 - Periodic Standardization report.
Available from http://www.seaclouds-project.eu/deliverables.html

[19] SeaClouds Deliverable D2.3.2 - Periodic Standardization report.
Available from http://www.seaclouds-project.eu/deliverables.html