



HAL
open science

Migrating Legacy Software to the Cloud with ARTIST

Alexander Bergmayr, Hugo Bruneliere, Javier Luis Cánovas Izquierdo, Jesus Gorrionogoitia, George Kousiouris, Dimosthenis Kyriazis, Philip Langer, Andreas Menychtas, Leire Orue-Echevarria, Clara Pezuela, et al.

► **To cite this version:**

Alexander Bergmayr, Hugo Bruneliere, Javier Luis Cánovas Izquierdo, Jesus Gorrionogoitia, George Kousiouris, et al.. Migrating Legacy Software to the Cloud with ARTIST. 17th European Conference on Software Maintenance and Reengineering (CSMR 2013), Mar 2013, Genova, Italy. 10.1109/CSMR.2013.73 . hal-00869268

HAL Id: hal-00869268

<https://inria.hal.science/hal-00869268>

Submitted on 11 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Migrating Legacy Software to the Cloud with ARTIST

Alexander Bergmayr^{*}, Hugo Brunelière[†], Javier Luis Cánovas Izquierdo[†], Jesús Gorroñoigoitia[‡],
George Kousiouris[§], Dimosthenis Kyriazis[§], Philip Langer^{*}, Andreas Menychtas[§],
Leire Orue-Echevarria[¶], Clara Pezuela[‡] and Manuel Wimmer^{*}

^{*}Vienna University of Technology, Austria

Email: [bergmayr|langer|wimmer]@big.tuwien.ac.at

[†]AtlanMod Team (Inria, Ecole des Mines de Nantes & LINA), France

Email: [hugo.bruneliere|javier.canovas]@inria.fr

[‡]Atos Spain S.A., Spain

Email: [jesus.gorronoigoitia|clara.pezuela]@atos.net

[§]National Technical University of Athens, Greece

Email: [gkousiou|dimos|ameny]@mail.ntua.gr

[¶]TECNALIA, ICT - European Software Institute Division, Spain

Email: Leire.Orue-Echevarria@tecnalia.com

Abstract—As cloud computing allows improving the quality of software and aims at reducing costs of operating software, more and more software is delivered as a service. However, moving from a software as a product strategy to delivering software as a service hosted in cloud environments is very ambitious. This is due to the fact that managing software modernization is still a major challenge; especially when paradigm shifts, such as moving to cloud environments, are targeted that imply fundamental changes to how software is modernized, delivered, and sold. Thus, in addition to technical aspects, business aspects need also to be considered.

ARTIST proposes a comprehensive software modernization approach covering business and technical aspects. In particular, ARTIST employs Model-Driven Engineering (MDE) techniques to automate the reverse engineering of legacy software and forward engineering of cloud-based software in a way that modernized software truly benefits from targeted cloud environments. Therewith, ARTIST aims at reducing the risks, time, and costs of software modernization and lowers the barriers to exploit cloud computing capabilities and new business models.

Keywords-Cloud Computing; Software Migration; Model-driven Engineering; Model-driven Reverse Engineering; Software Modernization

I. INTRODUCTION

With the emergence of cloud computing, Software as a Service (SaaS) is becoming mainstream. Hereby, software (or parts of it) is hosted in cloud environments [1] and consumable by globally distributed clients. Since cloud computing aims at improving the quality of delivered services concerning rapid elasticity and high availability, as well as at reducing costs of software operation by a ‘pay-as-you-go’ pricing model, there is an increasing need to move legacy software into the cloud of services.

However, the systematic and efficient modernization of legacy software to exploit current cloud-based technologies remains a major challenge. This is not surprising because

managing software modernization is still a significant challenge in general [2] and even more ambitious when a change of the software delivery paradigm needs to be addressed. Such a paradigm shift implies fundamental changes to how software is modernized, delivered and sold. Consequently, in addition to technical aspects, business aspects need also to be considered in the software modernization process.

At the same time, Model-Driven Engineering (MDE) [3] is emerging that emphasizes on the use of models and model transformations to raise the level of abstraction and automation in the development of software [4]. However, MDE techniques are not only relevant to develop new software from scratch, but also to modernize legacy software. In Model-Driven (Software) Modernization (MDM), models representing legacy software are (i) (semi-)automatically discovered in a reverse engineering step and (ii) transformed until the new software satisfies the modernization requirements in a forward engineering step.

The ARTIST project proposes a comprehensive MDM approach for migrating legacy software into cloud-based software that covers both technical and business aspects. MDE allows automating the various steps involved in the software migration, notably reverse engineering of legacy software and forward engineering towards cloud environments. Moreover, the proposed approach addresses explicitly a pre- and post-migration phase. This means, both the analysis of consequences of envisioned migration strategies and the evaluation of modernized software against predefined migration goals are considered.

In Section II we give an overall view of ARTIST by presenting facts and expected outcomes. Main challenges and achievements of ARTIST are addressed by Section III while the plan to evaluate these achievements is covered by Section IV. Projects closely related to ARTIST are presented in Section V. The paper is concluded in Section VI.

II. PROJECT FACTS AND OUTCOME

The *Advanced Software-based Service Provisioning and Migration of Legacy Software* (ARTIST) project (www.artist-project.eu) is coordinated by Clara Pezuela from ATOS, Spain. Ten partners establish the ARTIST consortium: ATOS (Spain), TECNALIA (Spain), Ecole de Mines de Nantes & INRIA (France), Fraunhofer (Germany), Vienna University of Technology (Austria), Engineering (Italy), Institute of Communication and Computer Systems (Greece), SparxSystems (Austria), Athens Technology Center (Greece) and Spikes (Belgium). The project duration is 36 months from October 2012 until September 2015 which means that ARTIST is still in an early stage. Total costs of the project are €9.690.538 with a funding of €6.953.705.

The major outcome of ARTIST covers a migration method and a comprehensive tool suite to support modernizing legacy software towards cloud-based technologies. ARTIST will build on an open and extensible development environment to realize a model-based reverse and forward engineering tool-chain with decision making, validation and testing capabilities for software modernization. Modernization artifacts are stored in a central repository to foster reusability across all modernization tasks and to establish a knowledge base of scenarios and best practices for software modernization. With respect to cloud-based technologies, ARTIST will elaborate performance stereotypes, and optimization and deployment patterns for software hosted in a cloud environment.

III. THE ARTIST PROJECT

ARTIST aims at developing methods, techniques, and tools that facilitate the modernization of legacy software by exploiting current cloud-based technologies. Modernizing software towards cloud-based technologies requires a migration comprising of substantial changes to the legacy software. Therefore, ARTIST proposes a model-based software *modernization process* (cf. Figure 1) that consists of a *pre-migration, migration, and post-migration* phase.

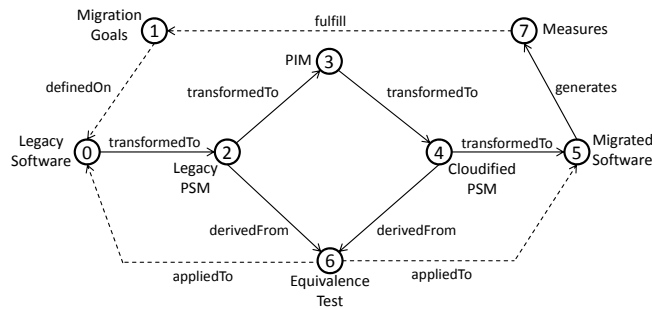


Figure 1. The ARTIST Software Modernization Process at a Glance.

Before the migration is performed, the *Legacy Software* ① is analyzed in the pre-migration phase concerning technical and non-technical consequences of possible migration

strategies. This analysis results in well-defined *Migration Goals* ① constituting the input for the decision-making on how the migration should be performed in the subsequent phases. In a first step of the migration phase, models are reverse-engineered from the legacy software. They comprise all specifics imposed by the platform the legacy software is built on. We refer to these models as *Legacy PSM* (Platform-Specific Model) ②. To enable the coverage of a wide range of current and future modernization scenarios and the reuse of reoccurring platform-independent migration patterns across several modernization scenarios, the legacy PSM is transformed into a higher-level representation, named *PIM* (Platform-Independent Model) ③. The PIM abstracts from platform-specifics, such as software runtime environments and data management capabilities. These platform-specifics need to be adapted to the offerings of cloud providers, as their cloud environments are typically unique and operate on different virtualization layers, i.e., from infrastructure to platform to software as a service. PIMs are then subjected to model transformations, which are selected based on the migration goals defined in the pre-migration phase. These transformations realize the actual migration by applying optimization patterns and integrating cloud-specific modernization opportunities. As a result, model-based representations of the migrated software that comprise platform-specifics compatible with the selected cloud environment are produced. Such a *Cloudified PSM* ④ is transformed into the executable *Migrated software* ⑤ hosted in a cloud environment. In the post-migration phase, the model-based representations of legacy software and migrated software are employed to derive *Equivalence Tests* ⑥. They aim at validating the behavioral equivalence of legacy software and migrated software. Furthermore, non-functional properties are evaluated to certify if the migration goals are fulfilled. This is achieved by analyzing the execution of migrated software to obtain quality *Measures* ⑦ which are checked against defined migration goals.

In the following, we discuss the major challenges addressed by ARTIST to cope with software modernization.

A. Planning and Assessment of the software modernization

In the pre-migration phase, the usefulness of the software modernization need to be clarified. This relates to the decision if the modernization of a legacy software is preferable over a new development from scratch provided that a cloud environment can satisfy the expectations and requirements of a software modernization at all from both a business and technical perspective. However, currently no well-established approaches exist to systematically predict the impact of a software modernization on existing business models, costs, and non-functional properties of the software, and to formally capture the goals of the migration in terms of a human-readable and machine-processable model.

ARTIST will address the business perspective by ana-

lyzing the impact of the business model imposed by cloud environments to existing business models and deriving necessary actions required to cope with potential organizational changes. Also the technical perspective needs to be considered in the terms of the impact of a cloud environment on the migrated software since cloud environments provide different capabilities to achieve improved software quality through, for instance, rapid elasticity, high availability, and enhanced performance. In ARTIST, current cloud environments will be examined based on the features they provide and the virtualization layers they address.

B. Operationalization of the software modernization

Once the decision to conduct the software modernization has been taken, an operational plan to accomplish the migration goals needs to be created. This includes gaining an understanding of heterogeneous legacy software artifacts, selecting an appropriate cloud environment for the software modernization, optimizing software artifacts in a way that they indeed exploit the capabilities of the selected cloud environment, and deploying the migrated software in the cloud of services. Since current approaches for software modernization still show limitations in terms of generating views on software artifacts that enable traceability, and automation [2], the migration of legacy software to cloud-based software remains a major challenge. Moreover, explicit descriptions for cloud environments and dedicated support to represent them are still missing, which hampers the generation of cloud-based software from legacy software.

ARTIST will facilitate a model-driven approach to conduct the migration by (i) (semi-)automatically discovering models from legacy software and creating comprehensive views, (ii) developing and applying model transformations to obtain platform-independent representations of the legacy software from which representations can be derived that are optimized to cloud-based capabilities, and (iii) generating source code that realize the migrated software. All software artifacts need to be correctly reconciled and integrated with cloud-specific features. Therefore, ARTIST will elaborate advanced model linking mechanisms among defined migration goals, model artifacts and transformations, and source code by providing dedicated repository capabilities. ARTIST will create model-based representations of features imposed by cloud environments and inject them into the developed model transformations as these representations determine the cloud-specifics built into the modernized software.

C. Validation and certification of the modernized software

In the post-migration phase, the migrated software needs to be validated to ensure that it behaves the same as the legacy software and fulfills the migration goals specified in the pre-migration phase. However, testing software equivalence in a feasible way constitutes a major challenge, especially when the test coverage is low for the legacy

software. Furthermore, certifying the quality of the migrated software is required to foster its adoption by clients since the users of ARTIST are turned from traditional software providers into service-based software providers. However, one major problem of new service-based software providers is the reluctance of their customers to consume new software offered as a service.

ARTIST will facilitate model-based testing by relying on the test cases migrated as part of the migration phase or by (semi-)automatically generating new test cases based on the model representations that describe the expected behavior. For validating the behavior not only component-level but also system-level tests are foreseen in ARTIST. Therefore, the legacy software and the migrated software will continuously be executed in parallel to compare their behavior. Finally, ARTIST will elaborate a certification model to demonstrate from a consumer perspective that the delivery of the migrated software in the cloud of services is similar or even improved in terms of non-functional requirements compared to the legacy software.

IV. PLANNED EVALUATION OF ARTIST

ARTIST will be evaluated by real-world case studies of ongoing commercial projects of the involved industry partners. The rationale behind the selection of the industrial cases is mainly based on these criteria: (i) variety of application domains such as e-government, media asset management, social networks and natural disaster management systems, (ii) computation and data intensive applications that show significant peaks in resource consumption, (iii) different software technologies including Java and .Net, (iv) desktop versus webtop client-server architecture, and (v) partial versus complete migration. The four ARTIST industrial cases can be summarized briefly as follows.

DEWS Case: The Distant Early Warning System (DEWS), developed by ATOS, is a distributed early warning disaster management system which requires real-time processing of information from a multi-sensor network to support decision making in natural disaster management. ARTIST will address the challenge of redesigning the architecture of the DEWS CCUI (Command-Control User Interface), from a desktop-based into a cloud-deployable webtop-based client-server application which should benefit from rapid elasticity, and ubiquitous and multi-device access.

SPCoop Case: The Public System for Cooperation (SPCoop) provided by Engineering supports its users to have a unified view on services of the Italian public administration. SPCoop integrates both the processes and data of different public administration services. ARTIST will improve SPCoop to benefit from the economies of cloud scale, enabling to offer common e-government services of SPCoop as a platform (PaaS) for the development of new software as a service.

NEWSASSET Case: NEWSASSET is a media asset management solution for news agencies, broadcasters or publishers marketed by ATC. NEWSASSET is a data intensive application managing news content such as images or video. ARTIST will address challenges with respect to cloud-based data management and highly geographically distributed data, and elasticity support for significant peaks in user requests.

UbiSoN Case: UbiSoN, developed by Spikes, is a social collaboration enablement facility targeting environments that have a rich real-time context. ARTIST will allow the application of UbiSoN as a social networking service to legacy software. UbiSoN serves as a testbed for partial migration of legacy software and its integration with existing services host in cloud environments.

V. RELATED PROJECTS

In the following, we discuss European FP7 projects that are closely related to ARTIST.

PaaSage (www.paasage.eu) is focused on cloud-based software development and run-time adjustments according to changing execution characteristics. Although ARTIST follows a similar line as PaaSage, the main motivation of ARTIST is software modernization which involves providing thorough representations of legacy software in terms of models. Model-based reverse engineering of legacy software and the planning of a migration to cloud environments is not in scope of PaaSage. Likewise in PaaSage, modeling plays a crucial role in ARTIST since modernization artifacts are represented on a model level from which the forward engineering is initiated. Run-time information is used in ARTIST for validating the migration success rather than execution adjustments as envisaged by PaaSage.

MODAClouds (www.modacLOUDS.eu) proposes, similar to ARTIST, a model-based migration approach. However, in MODAClouds the migration of cloud-based software between cloud providers and their interoperability is primarily focused rather than the migration of legacy software to cloud-based software as a means of software modernization. While in ARTIST a detailed model-based analysis of legacy software and migrated software is required, in MODAClouds a rather coarse-grained modeling approach seems to be applied for selecting cloud providers. Run-time information is employed in MODAClouds to monitor and reconfigure running cloud-based software. In ARTIST, we exploit run-time information in the post-migration phase. Such a post-migration phase is not foreseen in MODAClouds.

Legacy software migration in the cloud context is addressed by REMICS (www.remics.eu). They proposed PIM4Cloud, a metamodel for describing deployment artifacts. Such deployment artifacts are relevant also for ARTIST. However, to truly exploit the merits of cloud environments requires understanding of the structure, behavior and configuration of all artifacts involved in software

modernization that goes beyond the deployment viewpoint. In ARTIST, we advocate standard modeling languages such as UML to spot opportunities that allow building elastic software stacks distributed across cloud environments.

The focus of mOSAIC (www.mosaic-cloud.eu) is on cloud provider interoperability by supporting a search and negotiation process for service offerings. mOSAIC proposed an ontology for cloud provider characteristics which could be a basis for ARTIST to gain understanding of cloud environments. Generally, mOSAIC operates on a programming level while in ARTIST the modeling level is advocated.

Cloud4SOA (www.cloud4soa.eu) addresses, similar to mOSAIC, cloud provider interoperability by supporting search and adapter capabilities for service offerings. In Cloud4SOA, software migration between cloud providers is focused. In this sense, Cloud4SOA and MODAClouds share the idea of moving cloud-based software between cloud providers.

In summary, ARTIST distinguishes from existing projects by focusing on the migration of non-cloud software to cloud environments as a means of software modernization. ARTIST relies on current MDE techniques and explicitly deals with pre- and post-migration phases.

VI. CONCLUSION

Several obstacles exist today that hinder the modernization of software as a product to software as a service hosted in a cloud environment. ARTIST aims at establishing means to reverse engineer legacy software into models from which cloud-based software can be generated. To allow exploiting cloud-specific capabilities, ARTIST foresees several explicit actions for modernizing legacy software. With the methods, techniques, and tools developed by ARTIST, legacy systems are not simply moved into the cloud but enhanced in a way that they indeed exploit features enabled by cloud environments.

ACKNOWLEDGMENT

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

REFERENCES

- [1] M. L. Badger, T. Grance, R. Patt-Corner, and J. M. Voas, "Cloud Computing Synopsis and Recommendations," NIST Computer Security Division, Tech. Rep., 2012.
- [2] G. Canfora, M. D. Penta, and L. Cerulo, "Achievements and challenges in software reverse engineering," *Commun. ACM*, vol. 54, no. 4, pp. 142–151, 2011.
- [3] J. Bézivin, "On the Unification Power of Models," *SoSyM*, vol. 4, no. 2, pp. 171–188, 2005.
- [4] B. Selic, "MDA Manifestations," *UPGRADE, The European Journal for the Informatics Professional*, vol. 9, no. 2, pp. 12–16, 2008.