REMICS- REuse and Migration of legacy applications to Interoperable Cloud Services

REMICS Consortium
Andrey Sadovykh[1], Christian Hein[2], Brice Morin[3], Parastoo Mohagheghi[3], Arne J. Berre[3]

[1] SOFTEAM, France
[2] Fraunhofer FOKUS, Germany
[3] SINTEF, Norway
Contact persons: {Parastoo.Mohagheghi, Arne.J.Berre}@sintef.no

Abstract. The main objective of the REMICS project is to specify, develop and evaluate a tool-supported model-driven methodology for migrating legacy applications to interoperable service cloud platforms. The migration process consists of understanding the legacy system in terms of its architecture and functions, designing a new SOA application that provides the same or better functionality, and verifying and implementing the new application in the cloud. The demonstrations will cover the following REMICS research topics: model-based analysis and testing and model-driven interoperability with the tools by Fraunhofer FOKUS and SINTEF.

Keywords: Cloud computing, service-oriented architecture, legacy systems, ADM, model-based analysis, model-based testing, model-driven interoperablity.

1 REMICS Approach and Demonstrations

The REMICS[1] project will provide tools for model-driven migration of legacy systems to loosely coupled systems following a bottom up approach; from recovery of legacy system architecture (using OMG's ADM-Architecture Driven Modernization) to deployment in a cloud infrastructure which allows further evolution of the system in a forward engineering process. The migration process consists of understanding the legacy system in terms of its architecture, business processes and functions, designing a new Service-Oriented Architecture (SOA) application, and verifying and implementing the new application in the cloud. These methods will be complimented with generic "Design by Service Composition" methods providing developers with tools simplifying development by reusing the services and components available in the cloud.

[1] http://remics.eu/; funded by the European Commission (contract number 257793) within the 7th Framework Program

In order to instrument the migration process, the REMICS project will integrate a large set of metamodels and will propose several dedicated extensions. For the architecture recovery the REMICS will extend the KDM metamodel. On Platform Independent Model (PIM) level, the components and services are defined using SoaML (SOA Modeling Language[2]) which is developed in the SHAPE project[3]. The REMICS project will extend this language to address the specific architectural patterns and model driven methods for architecture migration, and to cover specificities of service clouds development paradigm. In particular, the PIM4Cloud Computing, model-driven Service Interoperability and Models@Runtime extensions are intended to support the REMICS methodology for service cloud architecture modeling. Furthermore, REMICS will investigate existing test notations such as the UML2 test profile (UTP) for their application to the SOA and Cloud Computing domain and refine and extend them.

The demonstrations focus on two aspects covered in the project: (1) the model-based analysis and testing and (2) model-driven interoperability.

The first demonstration shows two toolkits which are being developed within the REMICS. The toolkit consists of a model analysis workbench and a model-based testing workbench. The analysis workbench focuses on static aspects of all kind of engineering models. The workbench can be used to define and compute metrics for static analysis of the REMICS models. The measurement can be used to determine a certain degree of quality of the recovered and migrated models. In contrast to analysis workbench the model-based testing part is focusing on dynamic aspects of the models, for instance the generation of test specification and test data out of the behavioral description of the models is a typical activity with respect to model-based testing. These generated tests can be used by a test execution engine to validate specific behavioral properties of a system under test.

When the legacy system has been migrated to the cloud, it can leverage other services already deployed in the cloud to:

- Extend some of its services to provide some added value to migrated services
- Replace some of its services to use some similar services which provide better QoS, are cheaper to use, etc.

This leads to interoperability issues:

- How to exchange data between the migrated services and the “external” services available in the clouds?
- How to synchronize protocols (exchange of messages)?

In this second demo, we will focus on the first issue while the second issue is covered in the future research. Many tools (Semaphore, MapForce) already exist to define mappings between data-structures, and generate code out of these mappings to actually realize the mapping of data. However, some mappings are trivial, and explicitly specifying then is a waste of time. But some mappings are much more difficult to identify. Designers should thus be guided during this task. The tool to be presented in the demo provides guidance for designers to identify semantic mappings.

[2] http://www.omg.org/spec/SoaML/
[3] http://www.shape-project.eu/