Reuse and Migration of Legacy Systems to Interoperable Cloud Services- The REMICS project

Parastoo Mohagheghi, Arne J. Berre (SINTEF)
Project facts

- REMICS is a research project (STREP) accepted in the Objective 1.2 of FP7 Call 5 (Internet of Services, Software and virtualization).
- Aims at migration of legacy systems into Cloud services based on service-oriented architectures.
- REMICS runs from September 2010 for 3 years.
- The budget is 4.5 MEuro.
- Partners are:
  - SINTEF (Coordinator), DI Systemer (Norway)
  - Softeam, Netfective Technology (France)
  - Fraunhofer (Germany)
  - ESI, DOME Consulting and Solutions (Spain)
Problem to be addressed

- Legacy systems are sometimes of substantial value for companies:
  - They still function for the users’ needs;
  - They capture important business logic;
  - The cost of replacing them with systems designed from scratch is often too high.

- However:
  - Legacy systems are often difficult to reuse due to platform, documentation and architecture obsolescence.
  - New technologies arrive such as Cloud Computing and Software as a Service that promise better performance or cost saving that motivate organizations to modernize their applications.
Challenges

- The oldness degree of technologies to be reversed;
  - How to adapt them to the SaaS and cloud paradigms?
  - How to handle interoperability?
- The absence of knowledge;
  - How to extract business value information?
- QoS should be preserved;
  - How to reuse legacy systems in automated testing of the new SaaS?
- Cost of the migration process;
  - How to plan a progressive migration process?
  - How to train people in new technologies such as MDE?
Steps in the REMICS approach

1. Source Architecture
   - Recover
   - Legacy Artifacts

2. Migrate
   - Model Driven Interoperability
   - Target Architecture for Service Cloud platform
   - Forward MDA through PIM4Cloud

3. Validate, Control and Supervise
   - Service Cloud Implementation

4. SOA and Cloud Computing Patterns applied
   - Legacy Components Replacement and Wrapping
   - Design by Service Composition

Knowledge Discovery, Reverse Engineering

- Source code, binaries, documentation, users, knowledge, configuration files, execution logs and traces.

- RESERVOIR, Joyant, Amazon, Google, Microsoft

- Models@Runtime for application management
- Model Checking
- Model-based Testing for validation
Recover

- Analyse feasibility of the modernization strategies and select one;
- We plan to use the OMG KDM standard and extend it when necessary.
- Recover business value information: requirements, processes, rules, non-functional properties etc.
- Use automated reverse engineering methods as much as possible;
- Develop models (business, components, test specifications etc.) that will be used further.
Migrate

- The purpose is to start from the legacy models and refactor them to build a new SOA by applying methods such as decomposition, component wrapping and replacements.
- Some components or services may be replaced by newly discovered ones.
Compose and develop new services

- The legacy system may be enhanced by adding new services or services may be composed in new ways.

- Model-driven interoperability helps in adapting services using mediators. (Ref. Paper on Flora-2 interoperability mappings at MDI on Tuesday)

- Mediators or mediation services take input data in one format and provide it in another format.

- We plan to extend SoaML with data format models and behavioral model for mediation. (Ref. Talk on SoaML in Industry track on Tuesday)
Validate

- The recovered architecture should correspond to the legacy system and provide the same or better QoS, business goals, coverage, etc.
- Recovered models should be used in the validation process based on model-based testing techniques.
- The original system can act as a test oracle since requirements may not be well captured.
Control and supervise

- The goal is managing applications by observing them and performing corrective actions.
- Models@runtime for self-managability is one possible technique to use.
Technological approach: main points

- Model-driven techniques
  - Models everywhere,
  - A large set of metamodels and several dedicated extensions.
  - In particular, the PIM4 Cloud Computing, model-driven Service Interoperability and Models@Runtime extensions are intended to support the REMICS methodology for service cloud architecture modelling.

- Open source Metamodels and Models with an emphasis on Open Models for standards.

- Two pilot cases:
  - DI systems from Norway with ERP/accounting
  - DOME consulting from Spain within the tourism section
Expected impact

- REMICS will preserve and capitalize on the business value engraved in legacy systems to gain:
  - flexibility brought by Service Clouds,
  - lower the cost of service provision,
  - shorten the time-to-market.

- REMICS research will provide innovations in advanced model driven methodologies, methods and tools in Software as a Service engineering.

- REMICS will provide standards-based foundation service engineering and will provide a suite of open ready-to-use metamodels that lowers barriers for service providers.
REMICS and Standards

REMICS KDM Extension

CIM

BPMN
SBVR
BMM
ODM

PIM

SoaML

PIM 4 Cloud
PIM 4 Service Interoperability
PIM 4 Models @ Runtime

OCL
UML 2 Test Profile

REMICS PIM4ServiceClouds

PSM

UML Profiles for Web Services (XSD, WSDL, BPEL), OGSA and Service Cloud Platforms

Deployment Platforms

RESERVOIR, SUN, Joyant, Amazon, Google, Microsoft, Cordys
Extending SoaML from the SHAPE project

See SoaML in Industry track on Tuesday from 11-12
Thank you and

Questions?