Model-Driven Migration of Scientific Legacy Systems to Service-Oriented Architectures

Presentation at Model-Driven Software Migration (MDSM) 2011, Oldenburg, Germany, March 1st 2011

Jon Oldevik (SINTEF ICT)
Gørán K. Olsen (SINTEF ICT)
Ute Brönner (SINTEF M&C)
Nils Rune Bodsberg (SINTEF M&C)
Outline

- About oil spill modelling and our department
- Pilot application: MEMW/OSCAR
- Motivation: Modernizing a Fortran/C++ legacy app
- Approach overview
- Modelling: The migration profile
- Modelling: The PredictOilDrift pilot
- Generating code from models
- About data transformations
- Conclusion and outlook
Background: Oil spill modelling

- Oil spills can have disastrous effects on human safety, the environment, and businesses
- Understanding oil spill fate and effects is key to risk mitigation
- Much of SINTEF’s knowledge is manifest in numerical models
- These models are used for decision support
Activities at SINTEF M&C MET (Marine Environmental Technology)

- Laboratories
  - Oil chemistry
  - Analysis
  - Biology
- Meso-scale testing
  - Weathering and equipment testing
  - Sediments and shoreline
- Field studies
  - Oceanography
  - Surveillance
  - Releases
    - Experimental
    - Accidental
- Modelling
SINTEF M&C MET at a glance

- About 40 employees
- Turnover: 40-50 mnok - 90% petroleum industry, 10% governmental on average - 30% international portfolio
- Based in Trondheim, Norway

- Physics
- Mathematics
- Microbiology
- Ecotoxicology
- Statistics
- Informatics
- Analytical chemistry
- Oceanography
- Ecology
- Organic chemistry

Lab studies
Field trials
Meso-scale experiments
Numerical simulation
Oil spills – fate and effects

Volatilization from water to air

Surfacing of droplets, slick formation

Dissolution from entrained droplets
Degradation

Particulate adsorption/desorption
Settling

Sediments: deposition, dissolution, degradation

Surface processes:
- Drifting, spreading
- Natural dispersion
- Emulsification
- Evaporation
- Stranding

Response actions

Biological effects
Types of oil spills

- Accidental/regular
- Crude oil/refined products
- Surface/sub-surface
- Open sea/near coast/ice-infested waters

*Photo: Norwegian Coast Guard*
Oil spill-related decisions

Decisions on three levels:

- Strategic
- Tactical
- Operational

- Strategic and tactical decisions are based on historical (statistical) environmental data, whereas operational decisions are based on near real-time sensor or forecast data.

- Numerical models support decisions on all three levels.

“Should the Lofoten area be opened for oil production?”

“Where should oil booms for the Heidrun field be stored?”

“What do we do now? Apply dispersants, or use oil booms?”
What models do we have?

- Over decades, SINTEF M&C MET has developed a wide range of models.
- We have models (and sub-models) within the four sub-domains:
  - Oil weathering
  - Oil drift
  - Biological effects
  - Response options
How are our models implemented?

- Our models are engrained in applications (‘app trap’)
- These applications are used by oil companies and governmental bodies throughout the world
- Our main application is MEMW (Marine Environmental Modelling Workbench)
- MEMW contains the OSCAR (Oil Spill Contingency and Response) model
  - Oil weathering
  - Oil drift
  - Biological effects
  - Response options
- MEMW is a PC Windows applications written in Fortran and C++
MEMW/OSCAR screenshot (1)
MEMW/OSCAR screenshot (2)
Legacy architecture

- The OSCAR oil drift model runs within the PC/Windows application MEMW
- MEMW consists of a rich client (presentation layer) written in C++ and a FATES simulation engine (logic layer) developed in Fortran
- FATES is invoked when the user starts a simulation
- All input and output to/from FATES is file-based
- The rich client also handles data base access/file access for additional data (not from GUI)
Motivation: Modernizing a Fortran/C++ legacy app

- Current system (MEMW) is a commercial, monolithic C++ and Fortran application
  - Deployment and maintenance is per-customer
  - Integration with other systems and data is difficult
- Rationale for modernisation
  - Deployment and maintenance can be centralized
  - Facilitate integration with external systems
    - Service integration
  - Facilitate new business models and adaption to customer needs
- Why modelling?
  - Raises the abstraction-level
  - Provides an effective way of specifying services and integration needs
  - Can be used for code generation
Approach Overview

Step 1 (modelling)
- Migration profile
- Structured classes
- SoaML profile
- behaviour

Step 2 (code generation)
- Model 2 Text Transformations

Step 3 (deployment)
- Generated Web Service
  - Databases
  - Native Share libs
  - Executable components
  - Web services
  - External libraries
  - Data transformations
The UML migration profile represents semantics of different types of migration features. The semantics are used for code generation.

Component types represent different sorts of legacy components.
PredictOilDriftService – Use Cases

Use case diagram not used for code generation (only for documentation purposes)
Sequence diagram not used for code generation (only for documentation purposes)
The service interface

```plaintext
interface
PredictOilDriftInterface

garDefaultValues(): Property[*]
garOilTypes(): OilType[*]
garStatus(in String): PredictOilDriftStatus
'asynch' predictOilDriftAsync(in OilSpillInput)...
garPredictOilDriftResult(in String): OilDriftResult

eception
PredictOilDriftException

serviceInterface
PredictOilDriftService
```
Service and legacy components
Service de-composition
Data types used in the service(s)
Implementation of service operations in activity diagram

The activity diagram orchestrates calls to other components...
Transformations – code generation

- Transformations are written in MOFScript
  - Open source EMF-based model-to-text generator for Eclipse

- From this, we generate
  - Interfaces
  - Service implementation
  - Various legacy wrapper code
  - Behavior from activity diagram
  - (Also supports JPA mappings of entities)

```text
import "WebServiceGen.m2t"
import "EntityGen.m2t"
import "PersistenceManager.m2t"
import "stdlib.m2t"

texttransformation uml2service (in uml:"http://www.eclipse.org/uml2/3.0.0/UML") {

  var model:uml.Model = uml;

  uml.Type::main () {
    usePersistence = true;
    var entPack:uml.Package = null;

    if (self.isService()) {
      self.startGen();
    }
    ...
  }

  -----

  uml.Class::genClassFile(forwardingClass : uml.Class) {
    ' :
    /**
     * Class ' self.name ' generated by MOFScript, ' time()' ', ' date()'
     */

    public class ' self.name ' '{
      // declarations
      self.genOtherDecl();
      self.genPropertyDecl();
      ....
```
Web client → Web service → Legacy Simulation

Location:
- Scenario name: Skagerak
- Start: 15.1.2011, 12:39
- Period: 12
- Amount: 1
- Depth: 2
- Duration: 3
- OdType: AGULA
- Latitude: 58.66357331671117
- Longitude: 2.912817382812487

Predict Oildrift
Data transformations – done manually

- Existing data stored in binary files handled by legacy app
- Mapping these data automatically was a challenge
  - Many file formats; many data elements; complex data structures
  - Source and target layouts were quite different
  - So, these transformations were hand-coded
- We did experiment with a model-based approach
  - Where the data layout knowledge was embedded in a model, from which data readers and writers could be generated
  - This approach could be useful if there were many different data transformations using more or less the same data layout
Usages of migrated system

- "ENVISION" (Environmental Services Infrastructure with Ontologies):
  - EU FP7 research project to develop service composition platform

- "NorSpill":
  - Next-generation operational oil drift model for Norwegian met office
Conclusion and outlook

- We have presented a model-driven approach for legacy migration to service-oriented architectures
  - Black-box migration by wrapping legacy components using model-driven and generative techniques
  - Defined a UML profile for migration and a set of code generators for generating services and wrapper components
- We have tested the migration approach on an oil drift prediction system

- We have not addressed automation of legacy data mappings
- So far, we have only migrated one of the models in MEMW
  - Migrating other models should be “a walk in the park”
Acknowledgements

- SiSaS project:
  - "SINTEF Software As a Service"  
  - Internal strategic project within SINTEF that focusses on providing scientific software as a service  
  - Covers both existing and new software
Questions may also be submitted later to nilsrune.bodsberg@sintef.no
Thank you!