Model Driven Software Engineering for Grid Modeling, Optimization and Simulation

Marcello Vitaletti[1], Nicola Fontana[2], Maurizio Giugni[3] and Gianluca Sorgenti degli Uberti[4]
[1]*IBM Rome Software Laboratory, IBM Italia, Via Sciangai 53, 00144 Rome, Italy*
[2]*Department of Engineering, University of Sannio, Piazza Roma 21, Benevento, Italy*
[3]*Department of Hydraulic, Geotechnical and Environmental Engineering, University of Naples Federico II, Via Claudio 21, 80125 Naples, Italy*
[4]*ARIN S.p.A., Via Argine 929, 80147, Naples, Italy*

Keywords: Model Driven Engineering, Domain Specific Languages, Model Transformations, Smart Grids.

Abstract: A three-year research project about water grid technology ("WATERGRID research project") is led by ARIN – the company managing water distribution in the city of Naples – in a partnership with IBM and the University of Naples. Objectives of the project's initial phase include designing and prototyping a subsystem for grid modelling, optimization and simulation (GMOS). The GMOS subsystem implements state-of-the-art software kernels for the simulation of water distribution networks, including modules for the calibration of the hydraulic model and for an optimal partitioning of the grid. This paper illustrates general findings in applying model-driven software engineering to the architecture and design of the GMOS subsystem which largely abstract from the specific nature of the distribution grid as they could equally apply to the modelling, optimization and simulation of gas and electricity distribution networks.

1 INTRODUCTION

This paper illustrates general findings in applying model-driven software development technology to the architecture, design and implementation of a grid modeling, optimization and simulation (GMOS) subsystem in the context of a wider project (the "WATERGRID research project") aimed to an intelligent management of water distribution networks.

These findings largely abstract from the specific nature of the distribution grid as they would likely apply to the modelling, optimization and simulation of gas and electricity distribution networks.

Introducing computer simulations for managing and optimizing a smart grid generally comes with the following requirements:

- The grid topology used in simulations must reflect changes in the field which are stored in a geographical information system (GIS) or in enterprise asset management (EAM) systems.
- Parameter values characterizing the physical model of grid components (e.g. tanks, pipes, valves, pumps and turbines in a water grid) can be derived from asset/management data but often they must be the output of a calibration procedure.
- Automation and control logic installed on field devices must be reflected in a corresponding control model for use during the simulation.
- Initial conditions to be used for predictive simulations in a critical situation should be created from sensor data collected in real-time.
- Boundary conditions must be provided as well and they often include the estimated demand (of water, gas or electricity) at distribution nodes. Demands must be therefore obtained by statistically analysing time series of end-users consumption data.
- Computing the energy costs of grid operations requires a model of energy consumption and generation for different grid devices (e.g. for pumps and turbines in a water grid).

It follows from the preceding requirements that an effective use of computer simulations in the operation of smart grids needs several independent and complementary models. The following platform independent models are relevant for water grids:

- Grid Topology model;

324 Vitaletti M., Fontana N., Giugni M. and Sorgenti degli Uberti G..
Model Driven Software Engineering for Grid Modeling, Optimization and Simulation.
DOI: 10.5220/0004028003240327
In *Proceedings of the 7th International Conference on Software Paradigm Trends* (ICSOFT-2012), pages 324-327
ISBN: 978-989-8565-19-8
Copyright © 2012 SCITEPRESS (Science and Technology Publications, Lda.)

- Physical model;
- Automation/Control model;
- Initial Conditions model;
- Boundary Conditions model;
- Energy model.

In order to ascertain which collection of model instances a given simulation will be (or it was) based upon, model instances corresponding to the different parts (topology, physics, etc.) must have distinct identities (name, version). Model instances corresponding to a given part generally evolve independently from the other parts.

The global model instance being used for grid simulations in a specific context or at a given time is therefore the composition of several sub-domain model instances. As a most obvious example: the same grid topology and physical model instances may be employed in many simulations with different initial and boundary conditions.

The following section illustrates application scenarios from the "WATERGRID research project" providing motivations for the proposed modelling of the problem domain.

Software engineering aspects of GMOS development are discussed in the central section of this paper which deals with partial models, cross-references, model authoring and domain specific languages.

2 APPLICATION SCENARIOS

This section illustrates two use cases from the WATERGRID project providing motivations for the proposed modelling of the problem domain.

2.1 Calibration of the Hydraulic Model

Calibration is the process by which the hydraulic model parameters (generally the pipe roughness) are estimated by using the available sensor data. The simplest method is to calculate the covariance and the sensitivity matrix of the state parameters by assuming a preliminary estimate of the roughness, which can be assigned according to the pipe material and coating, age and water and soil characteristics. The optimal sensor location problem (sampling design) can be solved by running a simulation based on the current estimate of the hydraulic parameters and different boundary conditions e.g. corresponding to minimum, maximum and average demand patterns (Yu and Powell, 1994; Bush and Uber, 1998; Del Giudice and Di Cristo, 2003). Other approaches (i.e. shortest path algorithms) calculate the optimal location based on the network topology (de Schaetzen et al., 2000). In this case, no initial estimate of the parameters is required, and the sensor location can be obtained without running any simulation. Nevertheless, a certain objective function has to be maximized (or minimized) and optimization algorithms should be used. Data collected from the sensors are then used to find the best estimates for the unknown parameters (roughness coefficients). To this aim, optimization-based, explicit or implicit methods can be used (Savic et al., 2009).

In summary, the calibration scenario is one where simulations need to be repeatedly run for different purposes. Model instances representing the grid topology and initial conditions may be reused without changes in these simulations with boundary conditions chosen from a representative set of model instances. Finally, the physical model is typically updated at each step of an iterative process.

2.2 Running Predictive Simulations

In this scenario, the validated topology and physical models are used with newly created models for the initial and boundary conditions. The latter must be created from field sensor data in order to support reliable predictive simulations under new operating conditions. For example, this is necessary when analysing abnormal operating conditions which may occur in a water distribution system in case of break of transmission mains, for fire hydrant service, for simulating the travel times of a pollutant accidentally or intentionally injected into the system.

Moreover, predictive simulations can be used for design of extension and/or rehabilitation of existing water distribution networks.

3 PARTIAL MODELS, CROSS-REFERENCES, AUTHORING

Cross-referencing across models is required to combine the information contained in models of the physics, energy, initial and boundary conditions of grid objects with their topological relationships.

Simulation codes typically need to process all this information at once in each run. Therefore, one must always provide an exporter where cross-model references are resolved and information is produced in the format required for running the simulation. A similar situation arises with authoring tools, because different aspect of the same grid must be handled together in a given authoring sessions.

3.1 Late Binding of Cross-References

The adopted model partitioning strategy is best illustrated by considering a specific element of the problem domain – a junction – and examining how different attributes and relationships of this element are distributed across the modelling packages.

The diagram in Figure 1 shows classes from the topology model – in the yellow box – and from the physical and boundary-conditions models in the red and green boxes, respectively. The topology model captures the fact that a `Junction` *is a* grid `Node`.

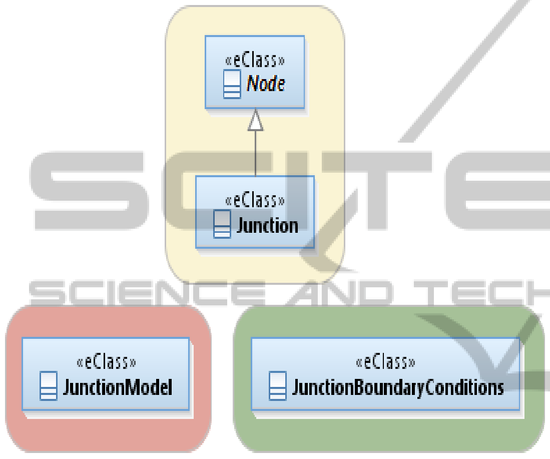

Figure 1: Modelling classes related to a grid junction.

The `Junction` class is not *extended* by classes at the diagram's bottom because the latter only capture specific complementary data. In the spirit of model partitioning they cannot incorporate the topological content expressed by the `Node` class.

With reference to Figure 1, it is desirable for an instance of say `JunctionModel` to provide a method returning a reference to the corresponding `Junction/Node` instance. The requirement can be satisfied by a form of late binding in which only a "name reference" (string) to the corresponding `Junction` object is included in `JunctionModel`.

While the non-topological `JunctionModel` cannot *extend* `Junction` there are situations in which it might be desirable for it to *implement* the `Node` interface, for example to make instances behave like ordinary nodes on a graphical map.

These situations are addressed by a variant of the well known Decorator pattern, where a class (`JunctionModel`) delegates the implementation of an additional interface (`Junction`) to an object natively implementing that interface. The solution is illustrated in Figure 2 for a junction grid object: at runtime, a reference to the `Junction` object is

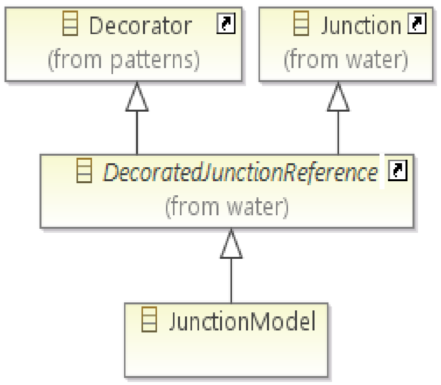

Figure 2: A variant of the Decorator pattern.

injected into the decorated name reference and used for delegating methods exposed by the topological (`Junction`) interface.

In summary, name references implement the necessary late binding between model objects capturing different dimensions of a domain element and the corresponding topology object (`Node` or `Link`). Additionally, name references may be decorated to make a non-topological object behave in a GUI like its node (or link) counterpart.

3.2 Model Authoring

Authoring tools are used to manually fill-in some missing data or to freely define all dimensions of an experimental test case. Dealing with multiple underlying model instances or files in the same logical session is impractical, therefore GMOS editors use a single-file representation of the information which would be otherwise distributed over instances of the partial models discussed in the previous section. This is obtained by introducing a an *authoring model* where all dimensions of a grid element are represented by a single class (logically an adapter). Transformations are also provided for:

- Merging model instances from the persistence layer into an instance of the authoring model; and
- Exporting an instance of the authoring model into instances of the partial models for storing changes in the persistence layer.

A wealth of EMF and related Eclipse tooling can be exploited with the GMOS domain models. In particular Graphical Editing Framework (GEF), the Graphical Modeling Framework (GMF) and Xtext are well known open-source frameworks by which it is possible and practical to develop domain specific languages (DSL).

3.2.1 Graphical DSL Editors in GMOS

A convenient approach implemented in GMOS separates graphical editing of the whole topology from (textual or graphical) editing of information related to grid elements (nodes or links). The spatial arrangement of grid objects (nodes) and connections (pipes) is rendered by adapting the grid topology model for display as a graph with GEF/GMF.

Data associated to all dimensions (including spatial data already captured by the graph editor) are presented in a "grid object editor". This is a form-based multi-tab editing panel with one field for each attribute of the corresponding grid object.

3.2.2 Text DSL Editors in GMOS

Text based editors extend the form-based object editor described in the previous section. In this case different text editors are defined, each one targeted to a specific object type (pump, valve, etc.) but covering all dimensions (including spatial data). A grammar for each editor is generated in Xtext from the corresponding object type in the authoring model. The following complication is inherent and worth noting in this approach: being the editor's scope a single object, the editor's target resource (text file) is only a temporary object whose content must be initialized from (and saved back to) the underlying instance of the authoring (global) model.

4 CONCLUSIONS

This paper illustrates some peculiarities emerging from the use of model driven software engineering in applications aimed to the modelling, simulation and optimization of water grids. The proposed approach is motivated by the requirement – likely applying to other smart grids – of supporting an independent life-cycle of model instances associated to different modelling dimensions. Significant impacts of this requirement were found in the areas of modelling, authoring and persistence.

REFERENCES

Steinberg, D., Budinsky F., Paternostro M., Merks E., 2008. EMF Eclipse Modeling Framework, *Addison Wesley*, 2nd edition.

Gronback, C. R., 2009. Eclipse Modeling Project – A Domain-Specific Language (DSL) Toolkit, *Addison Wesley*.

Yu, G., Powell, R. S., 1994. Optimal design of meter placement in water distribution systems, International *Journal of Systems Science* 25 (12), 2155–2166.

Bush, C. A., Uber, J. G., 1998. Sampling design methods for water distribution model calibration, *Journal of Water Resources Planning and Management* 124 (6), 334–344.

Del Giudice, G., Di Cristo, C. (2003). Sampling design for water distribution networks, *Proceedings of II Int. Conference on Water Resources Management*, Las Palmas, WIT-Press, pp.181-204

Del Giudice, G., Di Cristo, C. (2003). Nodal sensitivity and sensor location in hydraulic network, *Proceedings of XXX IAHR Congress - Tessalonica*, Theme B, pp.263-270

Savic, D. A., Kapelan, Z., Jonkergouw, P. M. R., 2009. Quo vadis water distribution model calibration?, *Urban Water Journal*, Volume 6, Issue 1, 3-22.

TRADEMARK STATEMENTS

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

IBM is a trademark of International Business Machines Corporation, registered in many jurisdictions worldwide.