Tool support for Collaborative Software Quality Management

Philipp Kalb and Ruth Breu

Institute of Computer Science
University of Innsbruck
Email: philipp.kalb, ruth.breu@uibk.ac.at

Abstract. Nowadays cloud services and complex cyber–physical systems gradually find their way into practice. As a result the need for end–to–end software quality management across platform and organizational boundaries has become paramount. One solution proposed by the software engineering community is the use of integrated model repositories for interchanging, interlinking and analyzing software engineering data and coordinating actions of manifold stakeholders working on this data. With MoVE, the Model Evolution Engine, we have developed a model repository supporting model–based data management in heterogeneous tool environments. The state machine based workflow concept allows a tight integration of data and automated and manual actions on the repository in a change–driven way. In this paper we will present the essential components of our MoVE Framework, starting with an introduction of the most important concepts, followed by the state based workflow language which will be contained in our demonstration [1].

1 Introduction

Modern software systems tend to consist of fragmented services across devices, platforms and organizational boundaries. To handle the rising complexity of such systems the consideration of end–to–end quality management is of major importance. For example the management of security in large–scale system like national health records requires coordinated efforts of heterogeneous stakeholders. Ranging from security engineers tackling technical issues such as designing secure software services to non–technical stakeholders such as compliance managers, surveying legal regulations, are also involved. As a consequence these systems demand for a consolidated treatment of data and processes in the realm of IT management, software engineering and systems operation [1].

Standards such as ITIL [2] and the software engineering community suggest the use of integrated model repositories for interchanging, interlinking and analyzing data [3,4,5]. While repositories have a long history in software engineering there exists still a huge gap in integrating different kinds of model–based data and semi–structured data. Additionally, the support of processes for end–to–end quality management, especially the interoperation of strictly structured

[1] `http://youtu.be/WKG__UnHL8U`

processes in IT management and agile processes in software engineering comes with further challenges. Flexible ticket based workflow management tools, such as IBMs Jazz platform [6] or Atlassians JIRA project management tool [7], have reached first adoption in practice in recent years. However, they do not address the aspect of data integration and are still weak in allying manual and automated tasks.

With MoVE, the Model Versioning and Evolution Engine, we have conceptualized and implemented a model repository referring not only to the data integration aspect but also the collaboration aspect. The MoVE–approach has a focus on continuous model integration for software engineering. MoVE provides methods to achieve traceability across tools, by applying concepts of meta–modelling and interlinkage. A key feature of MoVE is support for change–driven engineering, which is a novel methodology to cope with system evolution by supporting workflows triggered by changes of the systems data artifacts. The workflow language enables quality management to support change management as described in standards and guidelines such as ITIL [2] or ISO/IEC 20000 [8]. Hence, system evolution respecting data artifacts can be controlled to guaranty an integrated quality process during the complete software systems life cycle.

In Section 2 we will summarize the important concepts of the MoVE Framework. Section 3 describes the novel MoVE Workflow language, which is used to established a change–driven process.

2 Concepts and Architecture of the MoVE Framework

Figure 1 shows the overall architecture of the MoVE framework, consisting of the central MoVE Repository and multiple MoVE Clients connecting software engineering and IT management tools to the repository through MoVE Adapters.

From the conceptual point of view the basis of the MoVE Framework is the **Common Meta Model** (CMM). The CMM configures the data structures used in the MoVE Repository by specifying the (meta) model elements and their relationships. The CMM consists of a set of (partial) meta models such as the System Model, the Security Requirements Model, the Testing Model and the like. A full integration of all data structures of connected tools is not intended, the language should only contain the structures necessary for stakeholder collaboration. The CMM is designed using an UML modelling tool [2]. To enhance the UML models with MoVE specific features a UML profile (the MoVE Executeable Profile) defines a number of stereotypes. After its design the CMM is uploaded to the MoVE Repository with the help of a *Configuration–Service*. The Configuration Service uses the XMI representation of the CMM to configure the repository.

At the instance level Create, Read, Update, Delete and Query (**CRUDQ**) services the MoVE Repository provides to commit instances of the CMM. CRUDQ –Services are consumed by MoVE Adapters, which are plug–ins into client–side

[2] in the current implementation Magic Draw is used

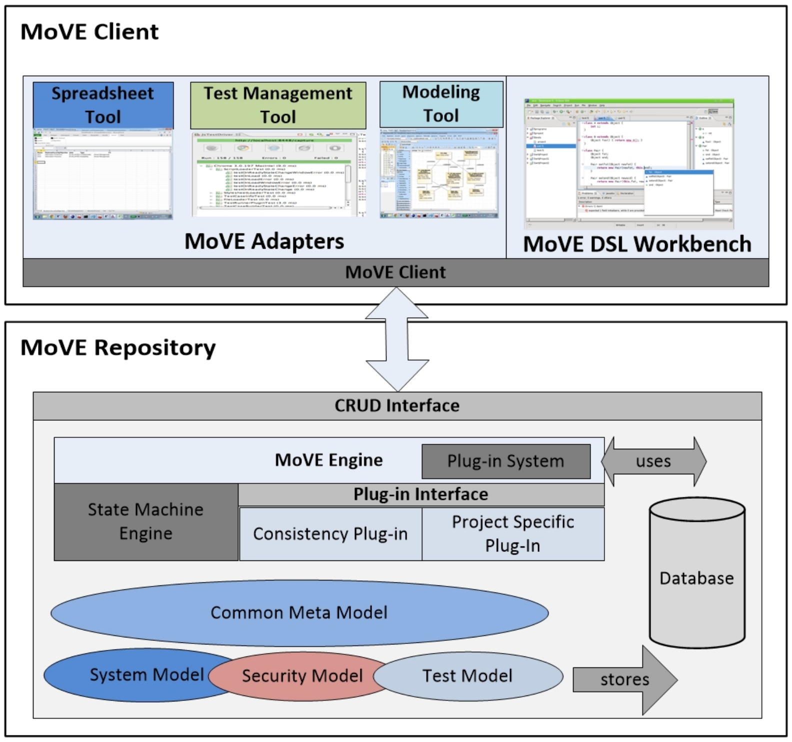


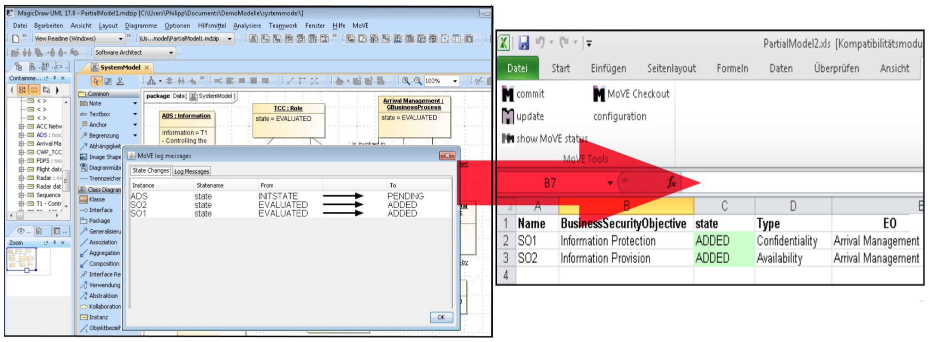
Fig. 1: Conceptual Architecture of the MoVE Framework

tools. These client–side tools are not limited to UML modelling tools. In the current environment we have developed e.g. a MoVE Adapter for Microsoft Excel to demonstrate the applicability of our concepts in a heterogeneous tool environment. The MoVE Adapter's main responsibility is to manage the mapping between the tool data representation and the representation in the MoVE Framework. CRUDQ–Services interact with the MoVE Engine, which is the main component of the MoVE server–side.

The major tasks of the MoVE Engine are to support versioning and persistency for all model elements stored in the MoVE Repository and to provide a **Plug–in Interface**. MoVE is event–driven in terms of generating an event for each change of a model or model element (using the CRUDQ–Services for changes). Each occurring change is analysed and then transformed into a change event. Server–side MoVE plug–ins listen to certain types of events and can trigger further actions. A **Plug–in System** allows users to register plug–ins for every (partial) model separately and therefore to decide which event should result in further actions. A crucial consumer of change events is the MoVE State Machine Engine, which allows to create state machine based workflows triggered by change events. Due to the importance of the MoVE workflow methodology it be described in Section 3 in more details.

3 The MoVE State Machine Workflow Language

The general idea of our state machine based workflow approach is that a model element can evolve during the operation of a system and typically undergoes a dedicated life cycle which is represented as a UML state machine.


MagicDraw UML 17.0 - PartialModel1.mdzip [C:\Users\Philipp\Documents\DemoModelle\systemmodel\]
Datei Bearbeiten Ansicht Layout Diagramme Optionen Hilfsmittel Analysiere Teamwork Fenster Hilfe MoVE
SystemModel
package Data[SystemModel]
ADS : Information
information = T1
- Controlling the
TCC : Role
state = EVALUATED
Arrival Management : GBusinessProcess
state = EVALUATED
MoVE log messages
State Changes
Log Messages
OK
PartialModel2.xls [Kompatibilitätsmodu
Datei Start Einfügen Seitenlayout Formeln Daten Überprüfen Ansicht
commit
update
show MoVE status
MoVE Checkout
configuration
MoVE Tools
B7

Instance	Statename	From		To
ADS	state	INITSTATE	→	PENDING
SO2	state	EVALUATED	→	ADDED
SO1	state	EVALUATED	→	ADDED

	A	B	C	D	E
1	Name	BusinessSecurityObjective	state	Type	EO
2	SO1	Information Protection	ADDED	Confidentiality	Arrival Management
3	SO2	Information Provision	ADDED	Availability	Arrival Management
4					

Fig. 2: System Model In Magic Draw with Updates for the Security Model in MS Excel

Each model element in the CMM can be attached with **states** and **state machines**. The states determine the quality gates in the quality lifecycle of the model element, like a *Security Requirement* being in the states *ADDED*, *COMPLETE* or *EVALUATED*. Transitions between states may be triggered in an *automatic* way by internal events stemming from other state machines, a timer or change events created by the MoVE Engine. Alternatively, *manual* transitions need user interaction which is implemented via systems such as mail [3]. Each transition can be guarded by conditions defined in OCL or the Hibernate Query Language (HQL).

Figure 2 shows two screenshots from our demonstration. The underlying CMM links a System Model with security requirements. The configured workflows control that on change of elements of the System Model, the linked security requirements have to be be re–evaluated. On the left side one can see a System Model designed with Magic Draw. On the right side, a spreedsheet–view in MS Excel contains the linked security requirements. Figure 2 shows the situation after an update of the System Model. The state of the linked security requirement changes from *EVALUATED* to *ADDED* and thus causes a re–evaluation of the security requirements.

In case a state has changed, it is possible to define a number of actions *onEntry* of the new state and *onExit* of the current state. These actions e.g. may involve external systems such as mail to notify stakeholders. Actions can be defined with two options: (i) The MoVE Executeable Profile contains several

[3] in our implementation we use Mylyn

predefined actions that can be composed in standard UML activity diagrams. Using this option it is possible to trigger predefined actions in a certain order but with limited expressiveness. (ii) Alternatively it is possible to use the fUML standard to create rich activity diagrams. FUML supports not only the actions defined in the MoVE Executeable Profile but a huge subset of UML activity diagrams. This enables users to create complex model changes on state changes.

4 Conclusion

In this demonstration paper we have sketched the MoVE framework which is a powerful model repository that integrates model storage capabilities.

The MoVE framework has been developed within the EU-IST project SecureChange [9] and has been employed as a central model repository for security policy interlinkage within the EU-IST project PoSecCo [10]. The has been evaluated in several case studies, both from applicability in industrial context, but also performance perspective. Within PoSecCo, the MoVE environment included six connected tools, using more than 4000 instances and about 15 state machines. The tool is available open source under Eclipse EPL license.

References

1. Breu, R., Agreiter, B., Farwick, M., Felderer, M., Hafner, M., Innerhofer-Oberperfler, F.: Living models - ten principles for change-driven software engineering. Int. J. Software and Informatics **5**(1-2) (2011) 267–290
2. APM Group Ltd: ITIL official website, accessed on february 19, 2014. http://www.itil-officialsite.com/home/home.aspx.
3. Sztipanovits, J.: Cyber physical systems - convergence of physical and information sciences. it - Information Technology **54**(6) (2012) 257–265
4. Atkinson, C., Stoll, D., Bostan, P.: Supporting view-based development through orthographic software modeling. In: ENASE. (2009) 71–86
5. Bruegge, B., Creighton, O., Helming, J., Kogel, M.: Unicase an ecosystem for unified software engineering research tools. In: Third IEEE International Conference on Global Software Engineering, ICGSE. (2008)
6. IBM: Jazz – rational team concert; project web side, accessed on february 20, 2014. https://jazz.net/products/rational-team-concert/.
7. Atlassian: Jira – project web side, accessed on march 20, 2014. https://www.atlassian.com/software/jira.
8. ISO/IEC: ISO/IEC20000. Information technology – Service management. ISO/IEC (2011)
9. SecureChange: EU project, accessed on june 30, 2014. http://www.securechange.eu/.
10. PoSecCo : EU project, accessed on june 30, 2014 http://www.posecco.eu/.