A METAMODEL INTEGRATION FOR METRICS AND PROCESSES CORRELATION

Xabier Larrucea and Eider Iturbe
European Software Institute Tecnalia, Parque tecnologico de Zamudio 204, 48170, Zamudio, Spain

Keywords: Processes and metrics definition and execution, Model driven engineering.

Abstract: Nowadays organizations need to improve their efficiency mainly due to the current economic situation. Several organizations are involved in process improvement initiatives in order to become more competitive in the market. These initiatives require processes definition and performance measurement activities. This paper describes briefly a metamodel integration between metrics metamodel and software and business execution metamodels in order to support this kind of improvement initiatives. In fact this integration implies to control coherently Software Metrics Metamodel for metrics, Software Process Engineering Metamodel 2.0 for defining processes and JBPM Process Definition Language for executing processes. This approach is supported by a prototype.

1 INTRODUCTION

Nowadays organizations need to improve their efficiency due to an increase of global competency in their markets. In fact they are starting up some improvement activities related to quality models. Specifically in the area of software quality and software development projects there is a great interest on process management. In fact this is related to one of the eight elements of software quality management (Humphrey, 2008)*: establish and maintain statistical control of the software engineering process*. In this sense and particularly in the software quality area we need not only define and execute software processes but also to provide a statistical control over them. Software and business processes are intertwined in many organizations but all of them are considered the organizations' engines. In this context metrics are the basis for a statistical process control and they are used to measure organization's behavior and performance (Florac et al., 1999). Several seminal papers have been published in the context of statistical process control (Oakland 2007), (Card, 1994) applied to software engineering.

Most of business process metrics are used to calculate the complexity, coupling, cohesion, size and modularity of business processes based on the elements used such as in (Mendling, 2009), (Vanderfeesten, 2007) and (Cardoso, 2007). Therefore business process metrics have been studied in for years and several metrics models has been applied to analyze their complexity. This is a similar situation with respect to software process metrics models where (Genero et al., 2005), (Garcia, 2004) and (Rolón Aguilar, 2006) are studies highlighting some metrics for software process models.

In the context of metamodelling a huge effort has been invested to develop metamodels for a wide range of domains. In terms of metamodelling our approach is compliant to the four architectural metalayers promoted by the Object Management Group (OMG). In our context some recent efforts have produced the Software Process Engineering Metamodel2.0 (SPEM, 2008) at OMG for describing software processes. In addition in the metrics knowledge area Software Metrics Metamodel (SMM, 2009) is developed for defining metrics.

Moreover in the area of execution languages there are several languages such as JBPM Process Definition Language (JPDL) and Business Process Execution Language (BPEL). Our approach is to combine all these metamodels in order to provide a coherent way to define and to model processes and metrics and afterwards to provide control over the execution of processes and their related metrics.

This paper is structured as follows. Firstly an overview of current approaches is highlighted.

Secondly an integration amongst metamodels is described. Thirdly an example of this approach is

Larrucea X. and Iturbe E. (2010).
A METAMODEL INTEGRATION FOR METRICS AND PROCESSES CORRELATION.
In *Proceedings of the 5th International Conference on Software and Data Technologies*, pages 63-68
DOI: 10.5220/0003039300630068

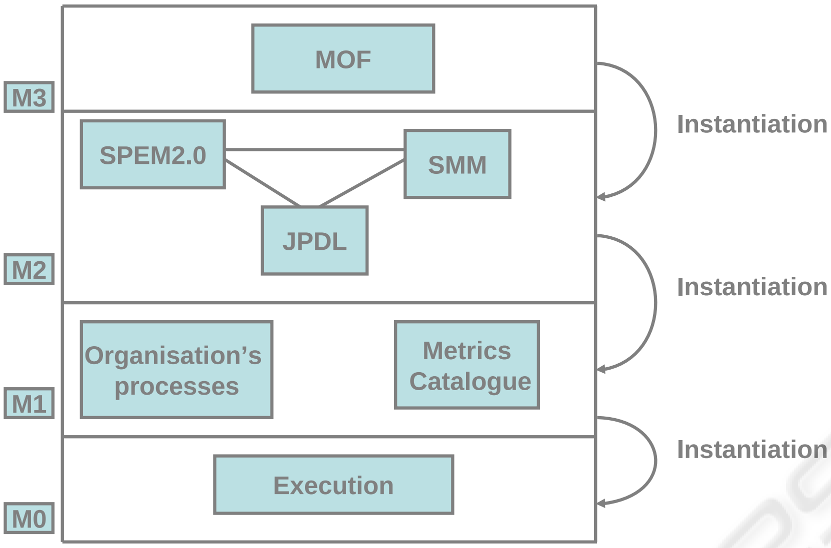

Figure 1: Model Driven Architecture Metamodels approach.

shown. And finally a conclusion section is provided in order to summarise the approach.

2 CURRENT APPROACHES

On the one hand metrics in software engineering has been widely studied such as in (Fenton et al, 1998) and (Kan 2004) where some metrics approaches are underpinned. (Ng Keng Yap et al. 2008) provide a robust data model for representing metrics.
On the other hand and as I mentioned before there are some approaches such as (Mendling, 2009) where author explores process models metrics such as complexity, coupling, cohesion, size and modularity. This approach is mainly focused on process models and their properties. Our approach is more similar to (Piattini et al. 2002), (Genero, 2005) and (Chatters et al. 1998) where the focus of the software quality is on measurements and metrics of processes performance. (Chatters et al. 1998) provide an example over a Cellular Manufacturing Process Model using predicting models for costs. (Piattini et al. 2002) outlines some steps for metrics definition and validation. In fact with respect to validation, authors identify theoretical validation and empirical validation. Our approach aims to overcome both approaches taken into account measurement theory-based approaches and its performance in empirical and real situations.
But there are more metrics applied to different aspect such as in (Genero, 2005) where the authors provide an overview of the existing proposals of metrics for conceptual metrics. These related works contribute to the set of metrics used in our approach. In fact our architecture is based on "the process virtual machine" provided by (Baeyens et al. 2010).

3 INTEGRATION METAMODELS

Our approach is basically based on the four-layer metamodelling architecture promoted by the OMG and its Model Driven Architecture.
Figure 1 summarises this approach where the highest metamodel is MOF (Meta Object Facility) (layer M3) and it is the basis for describing the subsequent metamodels through an instantiation process. Our proposal is to integrate SPEM2.0, SMM and JPDL metamodels coherently from a metamodelling perspective (layer M2).

Some parts of the used metamodels are not implemented because they are out of our scope. Figure 2 describes SMM metamodel elements implemented in our approach. We have extended this metamodel with the "variable" concept used for implementing and using different types for metrics measurements. Therefore we are able to define different types of measurements to processes and we have connected them to SPEM2.0 metamodel and JPDL metamodel (Figure 4). In addition "measurement" concept is related to "Element" that is the connection point with SPEM2.0 metamodel.

Figure 3 represents a snapshot of the SPEM2.0 metamodel where the main elements are "Process" and "TaskUse" representing processes and tasks elements used for the relationship with SMM. These elements are the cornerstone for this

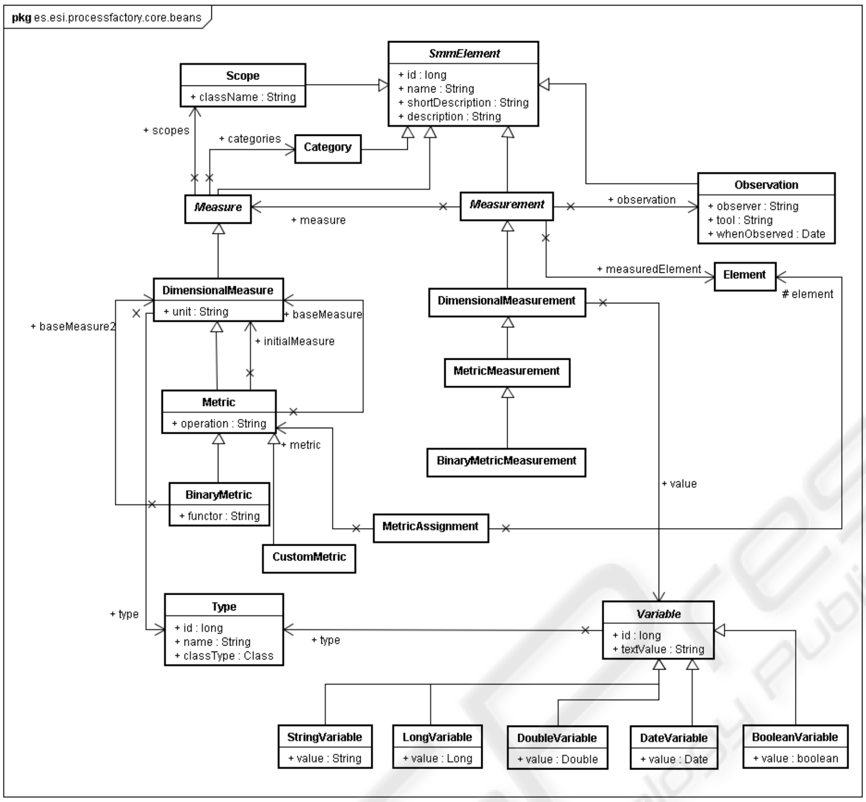

Figure 2: SMM metamodel elements.

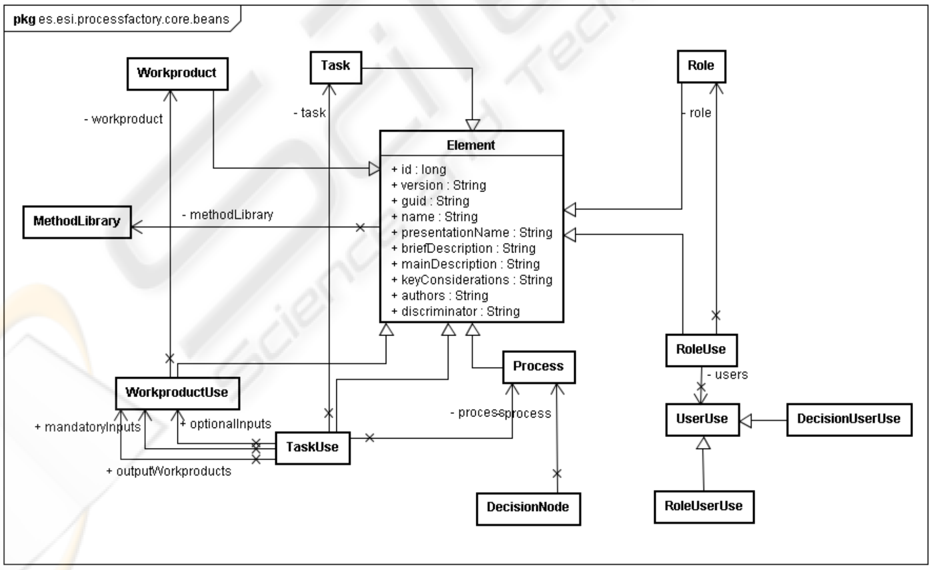

Figure 3: SPEM2.0 metamodel elements used in our approach.

integration because during definition and execution of metrics and processes we maintain a correlation between them.

As SPEM2.0 reuses UML activity diagram concepts for flow definitions, we have selected "fork", "join" and "decision" concepts as part of the behavior diagrams. All these elements are represented in JPDL concepts (Figure 4). This JPDL metamodel is only used for the final instantiation process (see Figure 1). This final step corresponds to

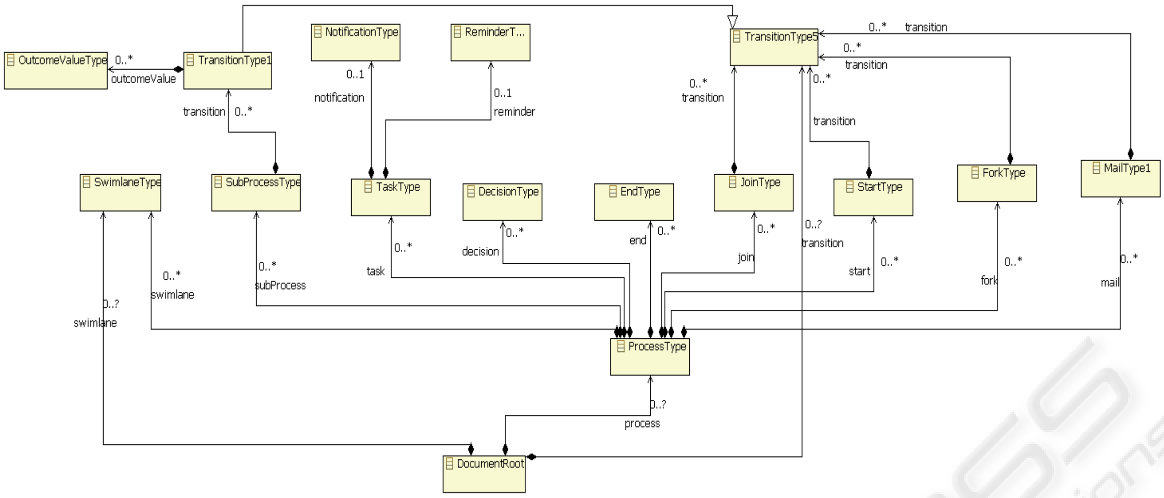

Figure 4: JPDL metamodel extracted from current XSD.

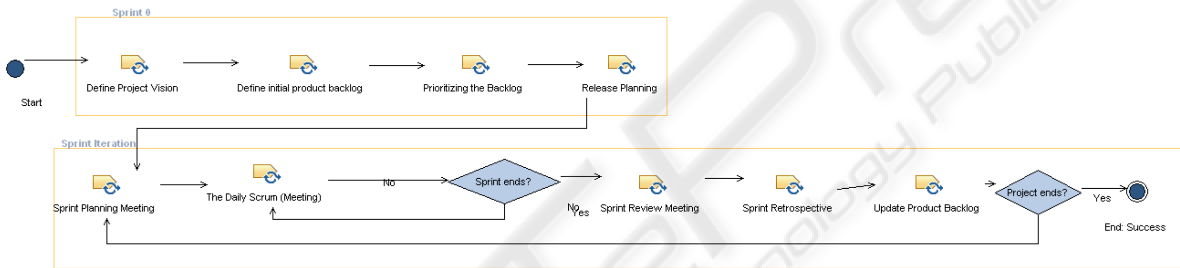

Figure 5: A scrum process modeled using EPF.

the process runtime and to its execution. In fact it is during process runtime when the correlation between metrics and processes definition and their execution should be maintained.

This integration provides us the ability to instantiate processes for an organization and to define a metrics catalogue that can be applied to these processes. Moreover the resulting approach allows us to instantiate during application runtime user's defined metrics depending on user's requirements. These users' metrics can be linked to processes and tasks.

Once processes and metrics are defined there is a transformation between SPEM metamodel and JPDL metamodel in order to deploy organizations' processes into the resulting application.

4 USE CASE

We have applied our approach to a software development process based on a Scrum methodology/framework (Schwaber, 2002) for an own development.

Scrum is an agile process or framework for managing agile projects as described in [http://www.scrumforteamsystem.com/ProcessGuidance/v2/FAQ/FAQ.aspx#whatIsScrum]. In this context there are three main roles involved in a scrum project: Product owner, Sprint Master and Sprint team. The work to be done on a Scrum project is listed in the Product Backlog, which is a list of functional and non-functional requirements including estimation of efforts in order to turn each into deliverable product increments. In Scrum, projects progress via a series of month-long iterations called *sprints*. At the beginning of each sprint, the scrum team selects a subset of functionalities from the product backlog and they work on it during the sprint, synchronizing their tasks in daily scrum meetings.

In a scrum project there is a preparation phase called sprint 0 and it is performed before a sprint iterations cycle, where an initial prioritized product backlog is defined and an initial release plan is created.

Based on this context we have used SPEM2.0 and the Eclipse Process Framework to define this Scrum process (see Figure 5). In fact there are

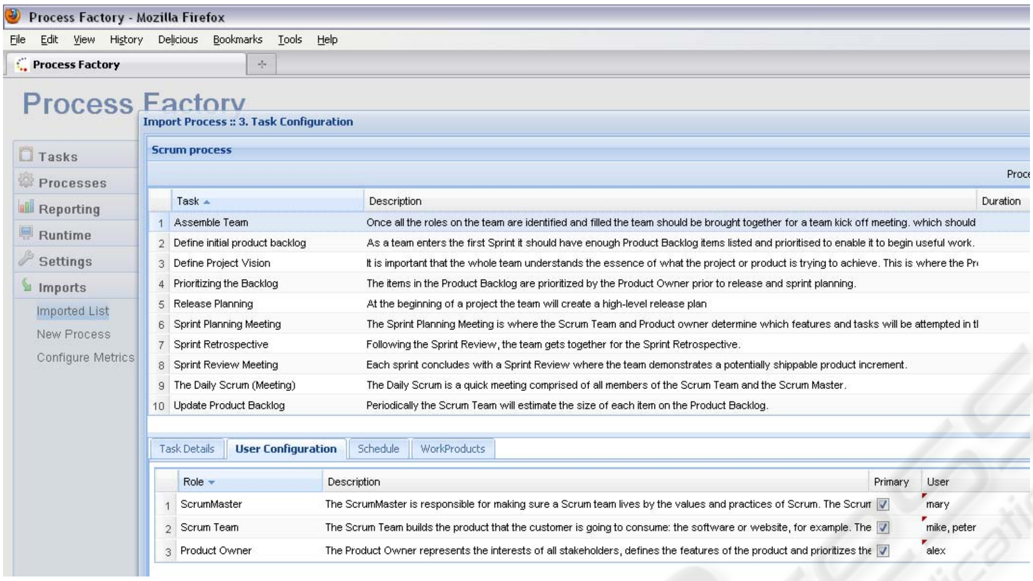

Figure 6: Process Factory interface during import.

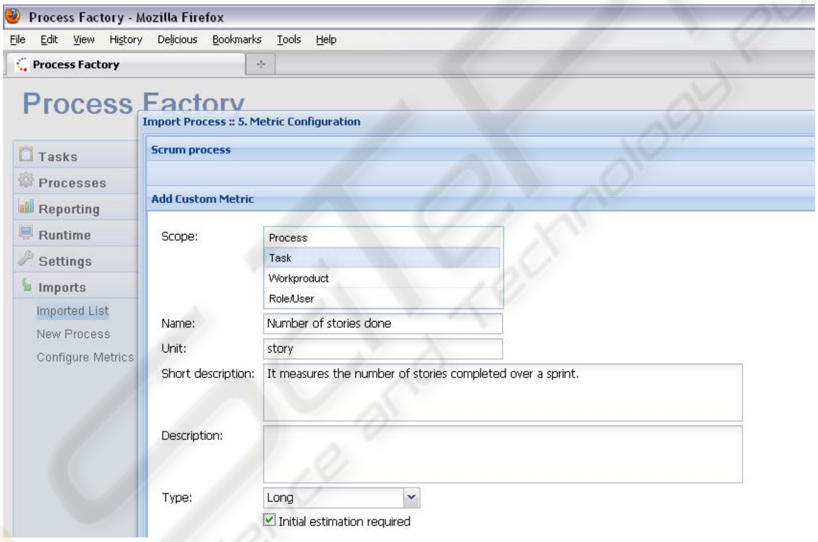

Figure 7: Adding a custom metric to a process.

several processes described using SPEM2.0 but it is not the scope of the paper to describe all these processes. Once we have defined this scrum process, we import the process and we transform a SPEM2.0 instance to a JPDL instance. The relationship between these elements is maintained through a database. At the same time we can assign metrics to this process from a metrics' catalogue or from a user definition. Some snapshots are provided in Annex (see Figure 6 and Figure 7). Thus we maintain integration not only at conceptual level but also at implementation level.

5 CONCLUSIONS

This position paper provides an approach to integrate processes and metrics from definition and execution points of view. This integration is not only at conceptual level that means we have extended all these metamodels and we have related them coherently. But also we provide a platform for definition and execution for metrics and processes where the connector elements maintain the correlation amongst them.

As future work we are planning to substitute

EPF tool by SAP gravity (http://www.sapweb20.com/) a web based tool for process modelling and some extension to open source tools such as ERPs.

We are working on a cloud based architecture supporting the full life cycle of this approach. In this sense we are testing our prototype in Google App Engine (GAE) (http://code.google.com/intl/en/appengine/).

ACKNOWLEDGEMENTS

This work has been supported by Vulcano project TSI-020301-2009-1 funded by the Spanish government (Ministerio de Industria, Turismo y Comercio) and FACIT-SME (FP7-2008-2-contract number: 243 695)

REFERENCES

Florac W., Carleton, A. “Measuring the software process: statistical process control for software process improvement”. *Addison-Wesley Longman Publishing Co.*, Inc. Boston, MA, USA Year of Publication: 1999 ISBN:0-201-60444-2

Oakland J.S. Statistical Process Control. Elsevier. ISBN-13: 978-0-7506-6962-7

Card D. “Statistical Process Control for Software?” *IEEE Software Volume* 11, Issue 3 (May 1994) Pages: 95 – 97. Year of Publication: 1994 ISSN:0740-7459

Mendling J. Metrics for Process Models. LNBIP 6. ISBN 978-3-540-89223-6. Springer Berlin Heidelberg, 2009, p. 103

Vanderfeesten I., Cardoso J, Mendling J, Reijers H. , Wil van der Alst. “Quality Metrics for Business Process Models”. In L. Fischer, editor, *BPM and Workflow Handbook* 2007, pages 179-190. Future Strategies Inc., Lighthouse Point, Florida, USA, 2007.

Cardoso, J. “Business Process Quality Metrics: Log-Based Complexity of Workflow Patterns”. On the Move to Meaningful Internet Systems 2007: *CoopIS, DOA, ODBASE, GADA, and IS LNCS Book* ISBN : 978-3-540-76846-3. Volume 4803/2010

Genero M., Piattini M., Calero C. “Metrics for software conceptual models”. Jan 2005 ISBN: 9781860946066

García F, Ruiz F, Piattini M. “Definition and Empirical Validation of Metrics for Software Process Models”. *PROFES* 2004: 146-158

Rolón Aguilar E, Ruiz F, García F, Piattini M. “Applying Software Metrics to evaluate Business Process Models”. *CLEI Electron*. J. (2006)

Baeyens T, Valdes Faura M. The Process Virtual Machine. http://docs.jboss.com/jbpm/pvm/article/ (last view 8th March 2010)

Software Metrics Metamodel. Object Management Group. http://www.omg.org/spec/SMM/1.0/Beta1/PDF/March 2009.

Software Process Engineering Metamodel 2.0. Object Management Group- http://www.omg.org/spec/SPEM/2.0/PDF/ 2008

Schwaber K, Beedle M. “Agile Software Development with Scrum”.Prentice Hall 2002. ISBN 0-13-067634-9

Ng Keng Yap, Abdul Azim Abdul Ghani, Ali Mamat, Hazura Zulzalil, The Robust Software Metric Data Model Defined in XML”, *IJCSNS International Journal of Computer Science and Network Security*, VOL.8 No.2, February 2008

Chatters, B, Hendersson P.Rostron C., “SIMMER: Software and Systems integration modeling metrics and risks” *EuroSPI 98*, Goteborg, November 1998

Piattini M , Genero M, Calero C. “Data Model Metrics”, Handbook *of Software Engineering and Knowledge Engineering* Vol II. 2002 ISBN: 981-02-4974-8

Kan Stephen H. “Metrics and Models in software quality engineering”. Addison Wesley. July 2004. Second edition. ISBN: 02019156.

Fenton N. Pfleeger S. L. Software Metrics: A Rigorous and Practical Approach, 2nd edition. 1998 ISBN:0534954251 *PWS Publishing Co*