KNOWLEDGE MODELING OF AGILE PROCESSES IN HEALTHCARE SYSTEMS DEVELOPMENT

Michal Košinár

Department of Computer Science, VŠB – Technical University of Ostrava, Faculty of Electrical Engineering and Computer Science

Abstract

Requirements on healthcare software products are becoming more and more complicated and software systems of today are characterized by increasing complexity and size. Therefore, software systems can no longer be developed feasibly without the processes supported by appropriate methods. We propose a method for configuration and modification of agile processes behind healthcare products development based on gathered knowledge and formal modeling. Our approach allows to support and optimize the processes with formal methods of modeling and machine-learning based simulations.

Keywords

Healthcare information systems, software products, SCRUM, agile, OWL, TIL, PROLOG, formal systems, Petri Nets

INTRODUCTION

In this paper we introduce software engineering method that meets application desiderata specified as follows:

- Iterative, step-by-step development of software processes in software companies involved in healthcare and biomedicine information systems
- Formalization and modeling of process; we will be able to formally specify a software process, identify weak points and optimize/modify the process based on a model

We develop a new method and software tools as a part of our research on formal methods, support of software processes modeling, simulation and executing. These tools allow users to model the process with formal methods, simulate it and optimize the whole process model and/or its process execution. This paper focuses on the modeling of the software processes in companies with formal methods to achieve mathematically precise and optimized version of the process.

SOFWARE PROCESSES AND AGILE

Business processes represent the core of company behavior. They define activities which companies (their employees) perform to meet their customers' needs. As for the definition of a business process, we quote from Workflow Management Coalition [6]: "Business process is a set of one or more linked procedures or activities which collectively realize a business objective or policy goal, normally within the structure defining functional roles and relationships." A software process is also a kind of business process; it comes with its development and maintenance phases. Our research aims at healthcare and biomedicine information systems development processes. This can be divided into activities of different kind, as they are defined by the process life-cycle engineering [5]. The process life-cycle spiral includes these activities: Meta-modeling; Modeling, Analysis; Simulation; Redesign; Visualization; Prototyping, walk-through and performance support; Administration; Integration; Environment generation; Instantiation and enactment; Monitoring, recording and auditing; History capture and replay; Articulation; Evolution; Process asset management. [5].

A. Agile methods

Changes in software products are aspects that are anticipated in agile development, and change is what the modern business software development is all about even in critical systems in a field of healthcare and biomedicine systems.

Implementation of the changes required by software product features or customers' needs might not be easy but agile methods respond to these requirements a give us a solution. The basic paradigm of agile processes is

that there should be only the bare minimum of documentation [2] and the production of the actual executable software product as its primary goal. The 'agile' idea is to develop the simplest solution that could possibly work for the current feature [1].

B. SCRUM

SCRUM is an agile, lightweight and adaptable process for managing systems development. SCRUM concentrates on how the team members should function in order to produce the required system in a constantly changing business environment. [9]

SCRUM starts with the assumption that software development involves several variables that are likely to change during the project. These include requirements, resources, time frame and technology. SCRUM defines an empirical process control for ensuring that the current status of the project is visible [10].

It is an iterative process: a project developed under SCRUM is divided into sprints, which are the iterations of SCRUM. It is incremental: each sprint produces a working, shippable version of the software with additional functionality [9].

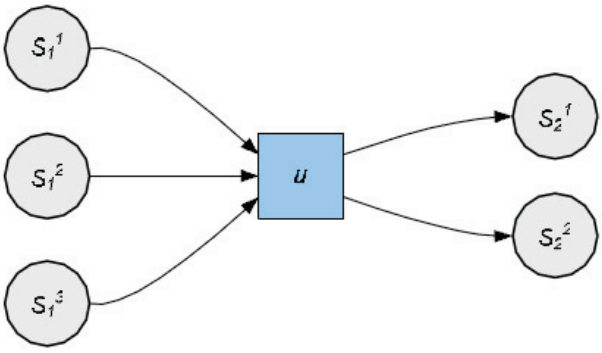

Fig. 1: Example of Petri net

FORMAL METHODS

Formal methods are techniques used to describe and model various systems as mathematical models. The big advantage of formal models we utilize in our research is the ability to verify the system's properties and simulate behavior of the system [11]. There are many formal languages that could be used for our needs like finite state automata, Petri nets or Workflow nets, or ontology languages like OWL, PROLOG or Transparent Intensional Logic (TIL).

We chose Petri Nets and TIL as a modeling languages as they meet our requirements for its transparency and rich procedural semantics and the ability of Petri Nets to be transformed to TIL.

C. Petri Nets

Petri nets are graphical and mathematical modeling tool applicable to many systems. A Petri net is a particular kind of directed graph, together with an initial state called the initial marking, M_0. The underlying graph N of a Petri net is a directed, weighted, bipartite graph consisting of two kinds nodes, called places and transitions, where arcs are either from a place to a transition or from a transition to place. In graphical representation places are drawn as circles, transitions as bars or boxes [3].

In modeling, using the concept conditions and events, places represent conditions and transitions represent events. A transition (an event) has a certain number of input and output places representing the pre-conditions and post-conditions of the event, retrospectively.

D. Transparent Intensional Logic

The standard modeling tools for ontologies is OWL [12]. However, it has some disadvantages; in particular using OWL we cannot define a multi-level classification, i.e. a class of classes, etc. (see [7]). Multilevel classification is also important during the use of meta-model based OMG specifications [13, 14]. For these reasons we prefer specification in TIL and we developed an OWL – TIL transformation (see [15]).

The key notion of TIL is that of a construction. It is an algorithmically structured procedure, or instruction on how to arrive at an output entity given some input entities. Constructions are assigned to expressions as their context-invariant structured meanings. There is an analogy between the notion of construction and that of a formula of formal calculi. What is encoded by a particular formula (or a λ-term of the TIL language) is just a construction of the (set of) model(s) of the formula. However, whereas formulas of a formal language are mere sequences of symbols that have to be interpreted in order to equip them with meaning, TIL constructions are just those meanings, i.e. procedures.

MODELING DISCIPLINE AND MODEL CREATION

We need to define the modeling discipline of processes in order to utilize formal models of software processes in simulation and optimization. The meaning of the term 'model' can be understood as a representation of one system – the modeled system – by another system. The modeled system often comes from reality or another artificial complex system and the model is its simplification – abstraction. Three functions of abstraction for modeling a process model are defined in [4, 19, 20]:

- Aggregation: an entity consisting of some other domain entities is modeled as one aggregated entity;
- Classification: a class of entities sharing some common features is identified;

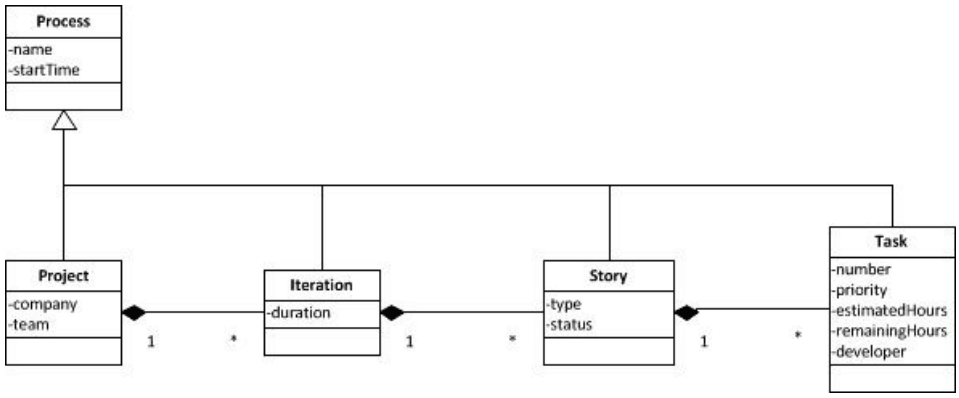

Fig. 2: Extract from Class Diagram of SCRUM Entities.

- Generalization: a new class of entities sharing common features is defined by abstracting from those features in which they differ.

However, the software-process development for healthcare and biomedicine domain has some specific features that must be taken into account (see [21]) and it has been characterized as "the most complex endeavor humankind has ever attempted" [8]. Nonetheless, a software process can and should be modeled in a formal way [17].

A knowledge-based approach to software process engineering has been dealt with in [18-20]. The benefits of the knowledge-based approach to modeling and simulation (KBS) in comparison with a Discrete-Event Simulation (DES - [21]) and System Dynamics (SD - [16]) are discussed in [12].

We have 3 basic types of entities that are fundamental for SCRUM process models: Project, Iteration, Task. A Project contains a set of Iterations. Iteration contains a set of Tasks. SCRUM follows an iterative approach to development, using time-boxed cycles. Each release of the system is implemented through a predefined number of time boxed Iterations. Iteration has a start time, duration and, again, contains a set of Tasks. Developers are the main actors/roles of an agile software development process.

DES generates experimental data and to facilitate the analysis of all this data, it is conventional to compress the data into a handful of meaningful statistics. In this paragraph are described two distributions from a family of continuous probability distributions. Exponential distribution refers to a statistical distribution used to model the time between independent events that happen at a constant average rate λ. Exponential distribution is used to model waiting time between events. The normal (Gaussian) distribution is a continuous probability distribution, and we have used it to generation duration of events (obstacles in development).

Proposed simplified agile model entities are (fig. 2)

- Project: This entity represents a software project. It is characterized by Name, a Start Time, a Company and a Team.
- Iteration: SCRUM follows an iterative approach to the development. It has its own identifier (Name), a Start Time, a Duration and contains a set of Stories.
- Story: Represents a single requirements specification of the system. It has its Name, a Type and a Status.
- Task: Each Task has name, a priority, an Estimated Hours, a Remaining Hours, and a Developer.

VALUE ADDED MODEL

Even though the software products in healthcare and biomedicine are based on similar and rather stable requirements the leading market healthcare and biomedicine companies often utilize SCRUM or other agile methodologies to support and control the development of their products. With more complicated software processes in a field of healthcare information systems development we can capture their fundamental and critical parts with formal systems and based on those we can optimize them and utilize such formal models further in validation, simulation and effort estimation.

Selection of healthcare and biomedicine domain was based on a strong knowledge behind medicine and hospital fields. With good formal process model and knowledge base behind software products development in software companies on the other side it will be much easier to gather requirements from customers and validate final software products.

CONCLUSION

In this paper we've introduced an approach to modeling of software processes for healthcare information systems development on agile method SCRUM and formal modeling tools. After the brief introduction we've described fundamentals of formal modeling, software process modeling discipline and creation and machine-learning and simulation methods.

Based on our research we introduced the approach we have implemented within the development of healthcare information systems for university hospitals and bigger clinics. The realization team of this software product

works under the SCRUM and with a good formal model and ontology described in a combination of PROLOG and TIL we could design and build a methodology and support tool for a development process simulation. These simulations can help the management to estimate the effort needed for the development of some product parts based on team attributes that are easy to get so they can optimize the process or plan better.

Future works should lead to more complex and precise formal model/ontology of the software process that is executed within the company and a simulation tool that will include more attributes and will be able to work with greater set of external factors that can influence the implementation of the software products. We are also working of our own modeling tool embedded into the simulation application so the tool will be more user friendly to end users.

ACKNOWLEDGEMENT

Michal Košinár is supported as a *Grand aided student of Municipality of Ostrava, Czech Republic*.

This research has been supported by the internal grant agency of VSB-TU of Ostrava - SP2013/207 „An utilization of artificial intelligence in knowledge mining from processes and process modeling and mining“.

REFERENCES

[1] K. Beck. Extreme Programming Explained: Embrace Change. Addison-Wesley, 2000. ISBN 0201616416.

[2] J. Highsmith and A. Cockburn. Agile software development: the business of innovation. Computer, 34(9):120–127, 2001.

[3] I. Vondrák. Methods of Business Modeling. VŠB-TUO, Czech Republic, Ostrava, 2004.

[4] M. Duží, M. Košinár, J. Kožusznik, S. Štolfa (2012) Knowledge-base approach to software process development based on TIL.

[5] Scacchi W, Mi P (1997) Process Life Cycle Engineering: A Knowledge-Based Approach and Environment. Intelligent Systems in Accounting, Finance, and Management 6:83--107-183--107.

[6] Workflow Management Coalition: http://www.wfmc.org/ WfMC Web, 1999.

[7] Machado EP, Caetano Traina J, Araujo MRB (2000) Classification Abstraction: An Intrinsic Element in Database Systems. Paper presented at the Proceedings of the First International Conference on Advances in Information Systems.

[8] Scacchi W (1999) Experience with software process simulation and modeling. Journal of Systems and Software 46 (2-3):183-192.

[9] K. Schwaber. Agile Project Management with Scrum. Microsoft Press, 2003. ISBN 0-7356-1993-X.

[10] Scrum. Control Chaos: Scrum. Internet, 2006. URL http://www.controlchaos.com/. Referenced 22.11.2006.

[11] Vondrák, I.: Neural networks (czech). VŠB-TUO, Czech Republic, Ostrava, revision 2009.

[12] W3C (2009) OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview/.

[13] Object Management Group(OMG) (2010) OMG Unified Modeling Language(OMG UML), Infrastructure, Version 2.3.

[14] Hug C, Front A, Rieu D, Henderson-Sellers B (2009) A method to build information systems engineering process metamodels. Journal of Systems and Software 82 (10):1730-1742. doi:10.1016/j.jss.2009.05.020

[15] Karkoška, T.: Vytvoření nástroje pro podporu tvorby ontologií v multi-agentním prostředí. Diploma thesis, VŠB-TUO, Czech Republic, Ostrava, 2008.

[16] Allemang D, Hendler J (2008) Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL.

[17] Laurent J-P, Ayel J, Thome F, Ziebelin D (1984) Comparative Evaluation of Three Expert System Development Tools: Kee, Knowledge Craft, Art. The Knowledge Engineering Review 1 (04):18-29. doi:doi:10.1017/S0269888900000631.

[18] Garg PK, Scacchi W (1989) ISHYS: Designing an Intelligent Software Hypertext System. IEEE Expert: Intelligent Systems and Their Applications 4 (3):52-63.

[19] Mi P, Scacchi W (1990) A Knowledge-Based Environment for Modeling and Simulating Software Engineering Processes. IEEE Trans on Knowl and Data Eng 2 (3):283-294. doi:10.1109/69.60792.

[20] Mi P, Scacchi W (1996) A meta-model for formulating knowledge-based models of software development. Decis Support Syst 17 (4):313-330. doi:http://dx.doi.org/10.1016/0167-9236(96)00007-3.

[21] Morgan Kaufmann. Silver GA, Lacy LW, Miller JA (2006) Ontology based representations of simulation models following the process interaction world view. Paper presented at the Proceedings of the 38th conference on Winter simulation, Monterey, California.

Michal Košinár
Department of Computer Science
VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
708 33 Ostrava – Poruba
Czech Republic
E-mail: michal.kosinar@vsb.cz