CAISE’2011 Forum

Preface

The CAISE conference theme is linked this year with the coming Olympic and Paralympic Games, which bring together athletes from all the continents to celebrate sporting excellence but also human diversity. Diversity is an important concept for modern information systems. Information Systems (IS) are diverse by nature, as well as the processes for constructing such systems, their developers, their users… It is therefore the responsibility of the IS Engineering community to engineer information systems that operate in such diverse world. During the two last decades, essential challenges made their appearance in the area of Information Systems related to engineering, quality and interconnectivity of information systems.

The CAiSE’11 Forum is a place within the CAiSE conference for presenting and discussing new ideas and tools related to information systems engineering. Intended to serve as an interactive platform, the forum aims at the presentation of fresh ideas, emerging new topics, controversial positions, as well as demonstration of innovative systems, tools and applications. The Forum session at the CAiSE conference will facilitate the interaction, discussion, and exchange of ideas among presenters and participants.

Two types of submissions have been invited to the Forum:

(1) Visionary short papers that present innovative research projects, which are still at a relatively early stage and do not necessarily include a full-scale validation.

(2) Demo papers describing innovative tools and prototypes that implement the results of research efforts. The tools and prototypes will be presented as demos in the Forum.

CAISE’11 Forum has received a record number of 46 submissions from 24 countries (Argentine, Australia, Austria, Brazil, Bulgaria, Canada, France, Germany, Hungary, Ireland, Israel, Italy, Japan, Latvia, Luxembourg, The Netherlands, Norway, Portugal, South Africa, Spain, Sweden, Switzerland, United Kingdom, United States of America). Among the submissions, 25 are demo papers and 21 are visionary papers.

The management of paper submission and reviews was supported by the EasyChair conference system. Selecting the papers to be accepted has been a worthwhile effort. All papers received three reviews from the members of the Program Committee and the Program Board. Eventually, 23 high quality papers have been selected; among them 16 demo papers and 7 visionary papers.

The CAISE’11 Pre-Proceedings available on this electronic support represent a collection of those 23 short research papers. Those papers included in the special proceedings issue titled “CAiSE’11 Forum” are published by CEUR.

After CAiSE’11, authors of the selected papers will be invited to submit an extended version of their papers for post-proceedings that will be published as a Springer LNBIP volume.

As the CAISE’11 Forum chair, I would like to express my gratitude to the Forum Program Board and the Program Committee for their efforts in providing very thorough evaluations of the submitted Forum papers. I wish also to thank all authors who submitted papers to the Forum for having shared their work with us.

Last but not least, I would like to thank the CAISE'11 Program Committee Chairs and the Local Organisation Committee for their support.

Paris, June 12th, 2011
Selmin Nurcan
CAISE'2011 Forum Chair

Program Board Members

Program Committee Members

Creating Declarative Process Models Using Test Driven Modeling Suite

Stefan Zugal, Jakob Pinggera, and Barbara Weber

University of Innsbruck, Austria
{stefan.zugal|jakob.pinggera|barbara.weber}@uibk.ac.at

Abstract. Declarative approaches to process modeling promise a high degree of flexibility. However, current declarative state-of-the-art modeling notations are, while sound on a technical level, hard to understand. To cater for this problem, in particular to improve the understandability of declarative process models as well as the communication between domain experts and model builders, Test Driven Modeling (TDM) has been proposed. In this tool paper we introduce Test Driven Modeling Suite (TDMS) which provides operational support for TDM. We show how TDMS realizes the concepts of TDM and how Cheetah Experimental Platform is used to make TDMS amenable for effective empirical research. Finally, we provide a brief example to illustrate how the adoption of TDMS brings out the intended positive effects of TDM for the creation of declarative process models.

Key words: Declarative Business Process Models, Test Driven Modeling, Test Driven Modeling Suite.

1 Introduction

In today's dynamic business environment the economic success of an enterprise depends on its ability to react to various changes like shifts in customer's attitudes or the introduction of new regulations and exceptional circumstances [1]. Process-Aware Information Systems (PAISs) offer a promising perspective on shaping this capability, resulting in growing interest to align information systems in a process-oriented way [2]. Yet, a critical success factor in applying PAISs is the possibility of flexibly dealing with process changes [1]. To address the need for flexible PAISs, competing paradigms enabling process changes and process flexibility have been developed, e.g., adaptive processes [3], declarative processes [4] and late binding and modeling [5].

Especially declarative processes have recently attracted the interest of researchers, as they promise a high degree of flexibility [4]. Although the benefits of declarative approaches seem rather evident [4], they are not widely adopted in practice yet. In particular, as pointed out in [4], [6] ,[7], understandability problems hamper the usage of declarative process models. An approach tackling these problems, the *Test Driven Modeling* (TDM) methodology, is presented in [7]. TDM aims at improving the understandability of declarative process models as well as the communication between domain experts [8] and model builders [8] by adopting the concept of *testcases* from software engineering. This tool paper describes *Test Driven Modeling Suite* (TDMS)[1] that provides operational support for TDM.

The remainder of this tool paper is structured as follows: Section 2 briefly introduces TDM. Then, Section 3 discusses the software architecture and features of TDMS, while Section 4 illustrates the usage of TDMS by an example. Finally, Section 5 concludes with a summary and an outlook.

2 Test Driven Modeling

In this section we briefly sketch what constitutes a declarative process model and how TDM is intended to support the creation of declarative process models. Please note that we focus on TDMS and necessary backgrounds only. A discussion of, e.g., related approaches, is out of scope and can be found in [7].

A declarative process model is characterized by a set of activities and a set of constraints. In contrast to imperative process modeling languages like, e.g., BPMN, the control-flow is not explicitly, but implicitly defined through constraints which exclude forbidden behavior. For instance, a constraint in process model S might specify that activity A is not allowed to be executed more than once. Then, every process instance that contains not more than one execution of A is considered to be a valid instance of S—independent of when A has been executed. An exemplary declarative process model can be found in Fig. 4 (2).

While constraints focus on forbidden behavior, TDM introduces the concept of *testcases* to focus on *desired* behavior of the process model. In particular, a testcase consists of an *execution trace* (i.e., a sequence of activities that constitute a process instance) as well as a set of *assertions* (i.e., conditions that must hold at a certain state of the process instance) (cf. Fig. 1). The execution trace of a testcase thereby specifies behavior that must be supported by the process model, whereas assertions additionally allow to test for unwanted behavior, i.e., behavior that must be prohibited by the process model. A typical example for an assertion would be to check whether activity N is executable at time M.

Consider, for illustration, the testcase depicted in Fig. 1. It contains the execution trace <A,B> (1) as well as an *execution assertion* that specifies that A cannot be executed between the completion of A and the start of B (2) and *termination assertions* that specify that the process instance *cannot* be terminated before the completion of A (3), however, it must be possible to terminate after the completion of A (4). The times in Fig. 1 do not necessarily constitute *real* times, but rather provide a timeline to test for control-flow behavior, i.e., define whether activities can be executed subsequently or in parallel. Furthermore testcases are validated automatically, i.e., no user interaction is required to check whether the specified behavior is supported by the process model.

So far we have introduced the concept of testcases, in the following we will sketch how their adoption intends to improve the communication between do-

[1] Freely available from: http://www.zugal.info/tdms

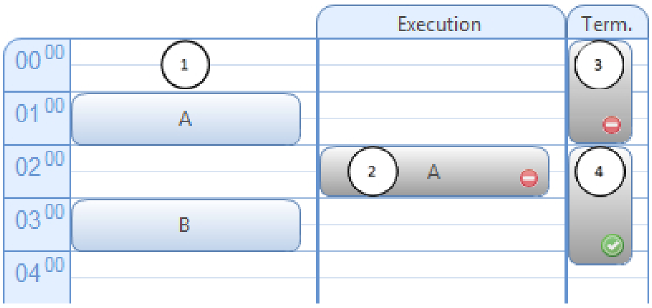


Fig. 1. A Simple Testcase

main expert (DE) and model builder (MB). Testcases provide information in a form that is not only understandable to the MB, but also understandable to the DE, who usually does not have the knowledge to read formal process models [8]. Usually the DE needs the MB to retrieve information from the model, cf. Fig. 2 (2) and (3). Since testcases are understandable to the DE, they provide an additional communication channel to the process model, cf. Fig. 2 (4) and (6). It is important to stress that TDM's intention is not to make the DE specify the testcases in isolation. Rather, testcases should be created by the DE and the MB together and provide a common basis for discussion.

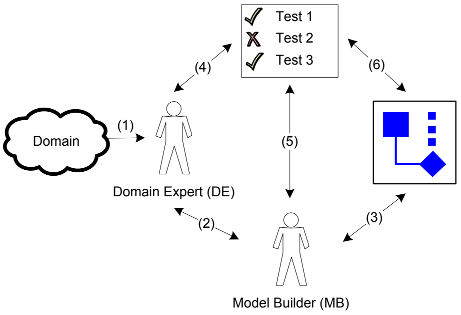


Fig. 2. Communication Flow

Besides improving the communication between DE and MB, testcases aim at improving the MB's understanding of the process model by providing an additional point of view. As pointed out in [7], especially so-called hidden dependencies [9], i.e., information that is not *explicitly* available in the process model can impede a model's understandability. An exemplary hidden dependency is shown in Fig. 3 (2): A must be executed exactly once (cf. cardinality constraint on A) and after A has been executed, B must be executed (cf. response constraint between A and B). Thus, B must be executed at least once for every process instance. However, this information is present in the process model implicitly only. Therefore the MB cannot rely on explicit information only, but has to inspect the model carefully for such hidden dependencies. Using TDM this problem can be tackled by specifying a testcase that tests for this hidden dependency as shown in Fig. 3 (1): the testcase specifies that the process instance can only be terminated if B has been executed at least once. As soon as the MB conducts changes to the process model that violate the testcase, the automated validation of TDMS (cf. Section 3) immediately informs the MB.

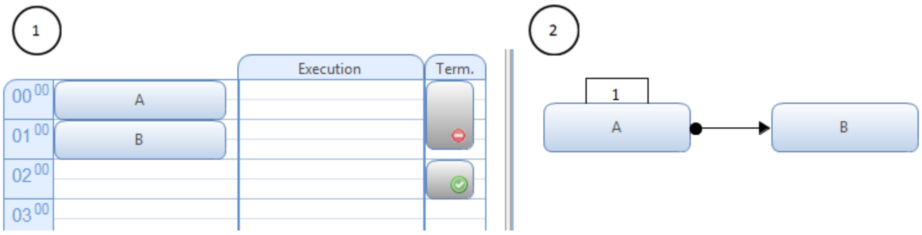


Fig. 3. Hidden Dependency

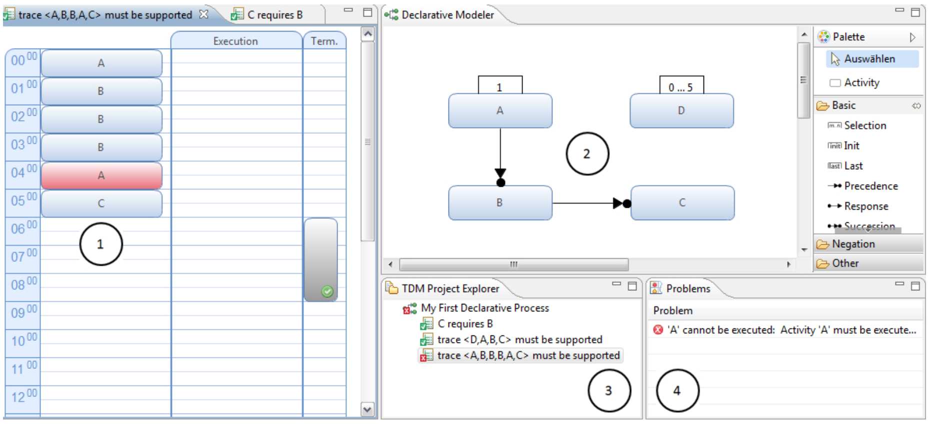
3 Test Driven Modeling Suite

Up to now we have introduced the concept of TDM. This section deals with Test Driven Modeling Suite (TDMS) which provides operational support for TDM. In particular, Section 3.1 discusses the features of TDMS in detail. Subsequently, Section 3.2 describes how TDMS is integrated with existing frameworks for empirical research and business process execution.

3.1 Software Components

To provide an overview of TDMS' features, all integrated components are illustrated in Fig. 4; each component will be described in detail in the following. On the left hand side TDMS provides a graphical editor for editing testcases (1). To the right, a graphical editor allows for designing the process model (2). Whenever changes are conducted, TDMS immediately validates the testcases against the process model and indicates failed testcases in the testcase overview (3)—currently listing three testcases from which one failed. In addition, TDMS provides a detailed problem message about failed testcases in (4). In this example, the MB defined that the trace $<A,B,B,B,A,C>$ must be supported by the process model. However, as A must be executed exactly once (cf. the cardinality constraint on A), the process model does not support this trace. In TDMS the failed testcase is indicated by the activity highlighted in (1), the testcases marked in (3) and the detailed error message in (4).

Testcase Editor. As mentioned before, testcases are a central concept of TDM, have precise semantics for the specification of behavior and still should be understandable to domain experts. To this end, TDMS provides a calendar-like testcase editor as shown in Fig. 4 (1).



My First Declarative Process
C requires B
trace <D,A,B,C> must be supported
trace <A,B,B,B,A,C> must be supported

Problem
'A' cannot be executed: Activity 'A' must be execute...

Fig. 4. Screenshot of TDMS

Declarative Process Model Editor. The declarative process model editor, as shown in Fig. 4 (2), provides a graphical editor for designing models in DecSerFlow [4], i.e., a declarative process modeling language.

Testcase Creation and Validation. In order to create new testcases or to delete existing ones, Fig. 4 (3) provides an outline of all testcases. Whenever a testcases is created, edited or deleted, or, on the other hand, the process model is changed, TDMS immediately validates all testcases and provides a detailed problem message in Fig. 4 (4) if a testcase failed. It is important to stress that the validation procedure is performed *automatically*, i.e, no user interaction is required to validate the testcases.

In order to ensure that all components work properly, TDMS has been developed using Test Driven Development, where applicable. In addition, researchers with different backgrounds, e.g., economics and computer sciences, have been included to develop an intuitive user interface. In a recent application of TDMS in a controlled experiment [10] no abnormal program behavior was observed. In addition, students considered TDMS as intuitive and easy to use.

3.2 Integration of Test Driven Modeling Suite

TDM, as introduced in Section 2, focuses on the modeling of declarative processes, TDMS provides the necessary operational support, i.e., tool support. To this end, TDMS makes use of Cheetah Experimental Platform's (CEP) [11] components for empirical research and integrates Declare [12] for workflow execution, as illustrated in Fig. 5 and detailed in the following.

Cheetah Experimental Platform as Basis. One of the design goals of TDMS was to make it amenable for empirical research, i.e., it should be easy to employ in experiments; data should be easy to collect and analyze. For this purpose, TDMS was implemented as an experimental workflow activity of CEP, allowing

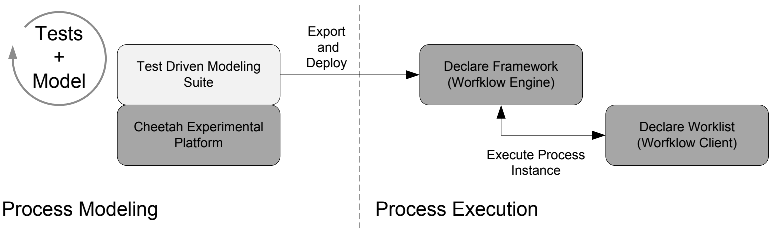

Tests
+
Model
Test Driven Modeling Suite
Cheetah Experimental Platform
Export and Deploy
Declare Framework (Worfklow Engine)
Declare Worklist (Worfklow Client)
Execute Process Instance
Process Modeling
Process Execution

Fig. 5. Interplay of TDMS, CEP and Declare

TDMS to be integrated in any experimental workflow (i.e., a sequence of activities performed during an experiment, cf. [11]). In addition, we use CEP to instrument TDMS, i.e., to log each relevant user interaction to a central data storage. This logging mechanism, in combination with CEP's replay feature, allows the researcher to inspect in detail how TDMS is used to create process models and testcases by watching the process of modeling step-by-step.
Business Process Execution. In order to allow for the execution of declarative process models created in TDMS, an export mechanism to Declare [12] is provided. As illustrated in Fig. 5, testcases and process models are iteratively created in TDMS. For deployment, the process model is converted into a format that can be directly fed into the Declare framework, i.e., workflow engine. Then, the Declare worklist allows for the execution of the process instance.

4 Example

A preliminary empirical evaluation shows the positive influence of TDM on cognitive load and perceived quality during model maintenance [10]. To illustrate the influence of TDMS on process modeling, we provide an example that shows how a DE and a MB could use TDMS to create a process model and respective testcases describing of how to supervise a master thesis (cf. Fig. 6–8). For the sake of brevity, the example is kept on an abstract level and the following abbreviations are used:
D: *Discuss topic* **P:** *Provide feedback* **G:** *Grade work*

Starting from an empty process model, the DE lines out general properties of the process: *"When supervising a master thesis, at first the topic needs to be discussed with the student. While the student works on his thesis, feedback may be provided at any time. Finally, the thesis needs to be graded."*. Thus, possibly with help of the MB, the DE inserts activities *D*, *P* and *G* in the testcase's execution trace (cf. Fig. 6); TDMS automatically creates respective activities in the process model. Now, the DE and MB run the testcase and the test engine reports that the testcase passes.

Subsequently, the DE and MB engage in a dialogue of questioning and answering [13]—the MB challenges the model: *"So every thesis must start by discussing the topic?"*. *"Yes, indeed—you need to establish common knowledge first."*, the DE replies. Thus, they create a new testcase capturing this requirement and run it. Apparently, the testcase fails as there are no constraints in the

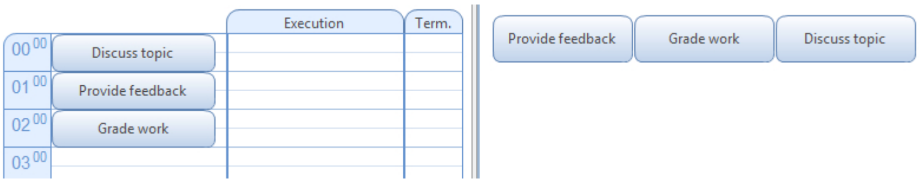


Fig. 6. Testcase 1: <*D*,*P*,*G*> Proposed by the DE

model yet. The MB inserts an init constraints on D (i.e., D must be the first activity in every process instance); now the testcase passes (cf. Fig. 7).

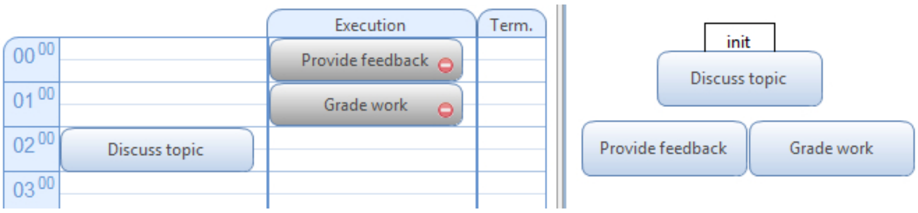


Fig. 7. Testcase 2: Introduction of Init on D

Again, the MB challenges the model and asks: *"Can the supervisor grade a thesis multiple times?"*. The DE replies: *"No, of course not, each thesis must be graded exactly once."* and together they specify a *third testcase* that ensures that G must be executed exactly once. By automatically validating this testcase, it becomes apparent that the current model allows G to be executed several times. Thus, the MB introduces a cardinality constraint on G (cf. Fig. 8).

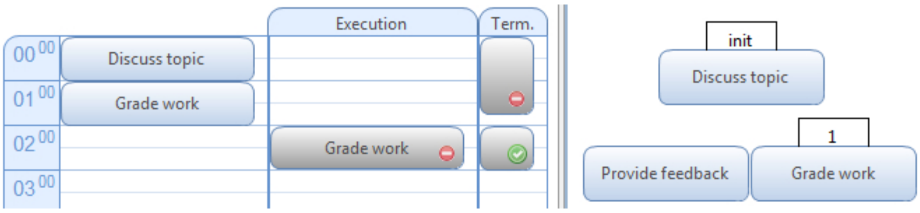


Fig. 8. Testcase 3: Introduction of Cardinality on G

While this example is kept small for the sake of brevity, it illustrates the benefits of using TDMS for modeling. First, the DE, who is usually not trained in reading or creating formal process models [8], is not required to modify the model itself, rather he defines behavior through the specification of testcases (possibly with the help of the MB). Second, testcases provide a common basis for understanding, thus supporting communication between the DE and MB. Third, behavior that is specified through testcases is validated automatically by TDMS, thereby ensuring that model changes do not violate desired behavior.

5 Summary and Outlook

TDMS, as described in this tool paper, provides operational support for the TDM methodology. More specifically, TDMS allows for a tight integration of declarative process models and testcases, thereby aiming at improving the communication between domain expert and model builder as well as resolving hidden dependencies. In addition, we sketched how we employ CEP as basis to make TDMS amenable for empirical research and showed how the Declare system is employed for the execution of declarative processes modeled in TDMS. Finally, we illustrated the intended usage of TDMS, in particular the iterative development of testcases and process model, with the help of a small example.

Future work focuses on further empirical validation: TDMS will be used in case studies to investigate whether the proposed methods are feasible in practice. In addition, TDMS will be employed in further controlled experiments to complement the case studies' results with quantitative data.

References

1. Lenz, R., Reichert, M.: IT support for healthcare processes - premises, challenges, perspectives. DKE **61** (2007) 39–58
2. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H.: Process Aware Information Systems: Bridging People and Software Through Process Technology. Wiley-Interscience (2005)
3. Reichert, M., Dadam, P.: ADEPTflex: Supporting Dynamic Changes of Workflow without Losing Control. JIIS **10** (1998) 93–129
4. Pesic, M.: Constraint-Based Workflow Management Systems: Shifting Control to Users. PhD thesis, TU Eindhoven (2008)
5. Sadiq, S.W., Orlowska, M.E., Sadiq, W.: Specification and validation of process constraints for flexible workflows. ISJ **30** (2005) 349–378
6. Weber, B., Reijers, H.A., Zugal, S., Wild, W.: The Declarative Approach to Business Process Execution: An Empirical Test. In: Proc. CAiSE '09. (2009) 270–285
7. Zugal, S., Pinggera, J., Weber, B.: Toward Enhanced Life-Cycle Support for Declarative Processes. JSME (accepted)
8. van Bommel, P., Hoppenbrouwers, S., Proper, E., van der Weide, T.: Exploring Modelling Strategies in a Meta-modelling Context. In: Proc. OTM '06. (2006) 1128–1137
9. Green, T.R., Petre, M.: Usability Analysis of Visual Programming Environments: A 'Cognitive Dimensions' Framework. JVLC **7** (1996) 131–174
10. Zugal, S., Pinggera, J., Weber, B.: The impact of testcases on the maintainability of declarative process models. In: Proc. BPMDS '11. (to appear)
11. Pinggera, J., Zugal, S., Weber, B.: Investigating the process of process modeling with cheetah experimental platform. In: Proc. ER-POIS '10. (2010) 13–18
12. Pesic, M., Schonenberg, H., van der Aalst, W.: DECLARE: Full Support for Loosely-Structured Processes. In: Proc. EDOC '07. (2007) 287–298
13. Hoppenbrouwers, S.J., Lindeman, L., Proper, E.H.: Capturing Modeling Processes - Towards the MoDial Modeling Laboratory. In: Proc. OTM '06. (2006) 1242–1252

A Semi-Automated Tool for Requirements Trade-off Analysis

Golnaz Elahi[1] and Eric Yu[2]

[1] Department of Computer Science, University of Toronto, Canada, M5S 1A4
gelahi@cs.toronto.edu
[2] Faculty of Information, University of Toronto, Canada, M5S 3G6
yu@ischool.utoronto.ca

Abstract. In designing most systems, requirements analysts face many competing requirements, such as performance, usability, costs, and so forth. Ideally, analysts would like to quantitatively measure consequences of solutions on requirements and risks, and extract stakeholders' preferences in terms of numerical weights. However, during the early stages of requirements and system design, it is hard to quantitatively measure all factors on a similar scale and quantify stakeholders' preferences. This contribution proposes a semi-automated decision aid tool which allows the use of available but potentially incomplete quantitative and qualitative requirements and risk measures. It removed the need to elicit importance weights of requirements. Instead, stakeholders are asked how much they would relax the demand on one objective to better achieve another. The proposed tool extends the Even Swap method with formally defined rules for suggesting the next swap to decision stakeholders.

Key words: Requirements trade-offs, qualitative decision analysis, preferences, quantitative data.

1 Introduction

Requirements analysts need to make key decisions early in the project, such as which architectural or design solution to employ [1]. Each alternative solution satisfies different functional and non-functional requirements to varying extents. Selecting a solution among multiple alternatives involves making trade-offs among requirements, with respect to stakeholders' preferences and consequences of alternatives on the requirements. Requirements analysts and project leaders also require objective risk measures to select good-enough countermeasures. In practice, however, quantifying risk factors, estimating probability and damage of risks, and quantifying the mitigating impacts of controls is challenging and error-prone [2].

Related Work: Faced with the typical absence of reliable quantitative data, some Requirements Engineering (RE) techniques, such as i^* [3] and Tropos [4] treat quality goals as soft goals. Goal model evaluation techniques such as [5–7], enable reasoning about the partial satisfaction of soft goals by propagating qualitative labels such as partially satisfied (✓.), sufficiently satisfied (✓), partially denied (✗.), and fully denied (✗). In some other RE approaches, requirements and alternatives are quantified by using ordinal measures or a probabilistic layer for reasoning about partial goal satisfaction [8–11].

Some decision analysis methods evaluate consequences of alternative solutions in terms of precise and meaningful quantitative measures [1, 8, 12]. Some Multi-Criteria Decision Analysis (MCDA) methods such as Analytical Hierarchy Process (AHP) [13] and Even Swaps [14], circumvent the need to measure requirements and consequences of solutions.

Problems: The main problems when making trade-offs among requirements to decide over alternative design solutions are:

1. Manual Prioritization: extracting stakeholders' preferences over multiple criteria in terms of numerical importance weights is error-prone and labor-intensive.
2. Incomparable Scales: aggregating requirements measures in different scales is usually error-prone or not possible.
3. Extensive Data Collection: eliciting required information to make an objective decision usually involves an extensive data collection from stakeholders.
4. Lack of Quantitative Risk Factors: quantitatively measuring the probability and damage of all risks is challenging, if possible at all.
5. Scalability: the decision problem may become complicated and impossible to be analyzed manually due to several requirements and/or alternatives.

Contributions: This paper describes a decision aid tool that addresses above problems. The tool adopts the Even Swaps multi-criteria decision analysis approach [14] to make trade-offs among requirements. The Even Swaps is a recently introduced decision analysis method in management science that consists of a chain of trading one decision criterion for another. These trades are called swapping. Swaps are even, which means stakeholders are asked to hypothetically improve one criterion, and in return, reduce another one proportionally (evenly). The main advantage of this method when dealing with software requirements is that it does not require extracting numerical importance weights and satisfaction level of all requirements on the same scale. Requirements can be evaluated in a mixture of scales and by different measurement methods.

Although the Even Swaps method solves the problem 1, 2, and 3 (mentioned above), it can fail in practice due to scalability issues: when several software requirements and alternative solutions need to be considered, decision stakeholders may not be able to determine the best swap among numerous possibilities [15]. The main contribution of this paper is introducing an algorithm (and tool) that:

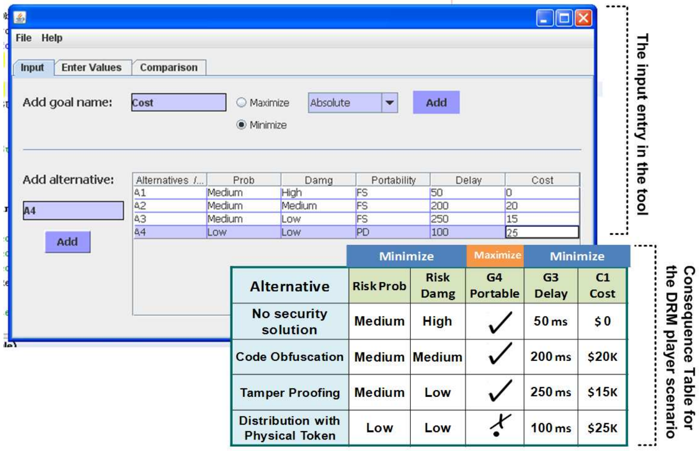
- Semi-automates the Even Swaps process, in the sense that the process is still interactive with stakeholders while being performed and controlled by an automated algorithm.
- The tool suggests which requirements to select for the next swap.

2 Motivating Example

We illustrate the use of the tool by analyzing a prototypical Digital Rights Managements (DRM) player [16]. The player gets an encrypted media and given a valid user key, decrypts the media, and decodes the digital content to an analogue audio. The user needs to purchase an activation code for the player, and the player only works if the activation code is valid at the time of using the player.

The DRM player contains two hard-coded credentials: *Valid activation code* and *Player key*. A software cracker can use the DRM player without buying activation code, either through static code analysis to extract the valid code or by tampering the binary code to bypass the license checks. The main protection strategies against *Tampering the binary code* attack are [16]:

1. *Obfuscation:* By obfuscating a program, the code is transformed into a form that is more difficult for an adversary to understand or change than the original code. Obfuscation adds an overhead to the code which causes performance drop downs.
2. *Tamper Proofing:* Tamper proofing algorithms detect that the program has been modified. Once tampering has been detected, a tamper response is executed which usually causes the program to fail.
3. *Distribution with Physical Token:* Physical tokens are hardware-based protections that try to provide a safe environment for data, code and execution. By employing a physical token, the user needs to show possession of a token to use the software.



Add goal name: Cost ○ Maximize ● Minimize Absolute

Add alternative: A4

Alternatives /...	Prob	Damg	Portability	Delay	Cost
A1	Medium	High	FS	50	0
A2	Medium	Medium	FS	200	20
A3	Medium	Low	FS	250	15
A4	Low	Low	PD	100	25

The input entry in the tool

	Minimize		Maximize	Minimize	
Alternative	Risk Prob	Risk Damg	G4 Portable	G3 Delay	C1 Cost
No security solution	Medium	High	✓	50 ms	$ 0
Code Obfuscation	Medium	Medium	✓	200 ms	$20K
Tamper Proofing	Medium	Low	✓	250 ms	$15K
Distribution with Physical Token	Low	Low	✗	100 ms	$25K

Consequence Table for the DRM player scenario

Fig. 1. Consequences of alternative security solutions on the DRM player requirements and the risk of tampering

These alternative security solutions have side-effects on other goals such as portability, delay, and cost. The corresponding consequence table in Figure 1 contains heterogeneous data, i.e., different goals are evaluated in different scales and by different techniques. Some of the criteria are measurable variables that need to be minimized or maximized. For example, stakeholders are able to estimate *Delay* (G_3) in milliseconds based on the properties and specification of alternatives. However, enough information is not available to quantitatively measure the risk of tampering attack, so consequences of alternatives on the damage and probability of this risk is evaluated in the ordinal scale of Low, Medium Low, Medium, Medium High, and High.

3 Basics of the Even Swaps Method

In an even swap, the decision analyst, collaborating with the stakeholders, hypothetically changes the consequence of an alternative on one requirement, and compensates this change with a preferentially equal change in the satisfaction level of another requirement. Swaps aim to either make criteria irrelevant, in the sense that both alternatives have equal consequences on the criteria, or create a dominant alternative. Alternative A dominates alternative B, if A is better than (or equal to) B on every criteria [15]. Irrelevant goals and dominated alternatives can both be eliminated, and the process continues until only the most preferred alternative remains [15].

Notation Remark. A swap between two goals g_x and g_y that changes the satisfaction value of g_x from x to x' and compensates this change by modifying the satisfaction level of g_y from y to y' is written as:

$$(g_x : x \rightarrow x' \Longleftrightarrow g_y : y \rightarrow y')$$

Case Study: The DRM Player Decision Scenario. We illustrate the Even Swaps method by analyzing and comparing the first two alternatives security solutions for the DRM player (Figure 1):

1. Compare *No security solution* (A_1) and *Code obfuscation* (A_2)
2. Ask stakeholders: If the $RiskDamg$ could be reduced from High to Medium, how much $Delay$ they would tolerate instead of 50 ms delay?
   $(RiskDamg : High \rightarrow Medium \Longleftrightarrow Delay : 50ms \rightarrow ?)$
   Stakeholders agree with increasing the Delay to 500 ms.
3. Consequences of A_1 are revised based on the above swap. The revised alternative is called A'_1, which is a virtual alternative and subsidence of A_1:
   Consequences of $A'_1 = \{Medium, Medium,\ \checkmark, 500ms,\ \$0\}$
   Consequences of $A_2 = \{Medium, Medium,\ \checkmark, 200ms,\ \$20K\}$
4. $RiskProb$, $RiskDamg$, and Portability are irrelevant decision criteria and can be removed:
   Consequences of $A'_1 = \{500ms,\ \$0\}$
   Consequences of $A_2 = \{200ms,\ \$20K\}$
   on $\{Delay, Cost\}$
5. Ask stakeholders: If the Delay could be reduced from 500 ms to 200 ms, what Cost they would pay?
   $(Delay : 250ms \rightarrow 200ms \Longleftrightarrow Cost : \$0 \rightarrow ?)$
   Stakeholders agree to pay \$10K for A_1.
6. Consequences of A'_1 are revised based on the above swap.
   Consequences of $A'_1 = \{200ms,\ \$10K\}$
   Consequences of $A_2 = \{200ms,\ \$20K\}$
   on $\{Delay, Cost\}$

7. Delay is irrelevant and is removed.
8. With respect to price, A'_1 dominates A_2.
9. A_2 is removed from the problem and the process continues by applying the Swap Method to A_1 and A_3.

4 The Automated Even Swaps Tool

This paper introduces a semi-automated Even Swaps tool, that given a consequence table, suggests a chain of swaps to determine the overall best alternative. The algorithm consists of several Even Swaps cycles. In the beginning of each cycle, the algorithm selects a pair of alternatives for the next Even Swaps process. The algorithm suggests a chain of swaps to stakeholders intending to find the preferred alternative in the pair. The dominant alternative is kept in the list of solutions, but the dominated alternative is removed. The algorithm then selects another pair of alternatives to compare and a new cycle starts. These cycles continue until one alternative remains, which is the best solution overall.

4.1 Automatically Suggesting Swaps

The decision aid algorithm has two main goals: 1) minimizing the number of swapping steps required to make one of the alternatives dominant, and 2) generating swaps that are easy to make for stakeholders. Toward these goals, we develop a set of rules in the following sections for suggesting the next swap to stakeholders.

Goal one: Minimizing the number of swapping steps: The tool minimizes the number of swapping steps in 2 ways: 1) suggests swaps that help make one of the alternatives dominant, and 2) reuses previously made swaps to avoid asking repetitive swap queries from stakeholders.

Rule 1: Make swaps that help toward creating a dominated alternative. Swaps make one of the decision criteria irrelevant, which helps remove one goal from the decision problem in each step. Removing criteria one by one is a time-consuming approach to apply the Even Swaps process. The tool suggests swaps that help toward making one of the alternatives dominated. That means if an alternative such as A is dominant for n goals like g_1, g_2, ... g_n, and B is a better solution only for one goal, like g, then we need to remove g by swapping it with one of those n goals. By removing g, solution A might still be dominant with respect to all goals (g_1, g_2, ... g_n). However, swapping any two goals from g_1, g_2, ... g_n will not change the situation between A and B and neither of them becomes dominated with respect to all relevant and remaining goals.

In the Even Swaps process, between the pair of A_1 and A_2 in the DRM player scenario, with respect to $RiskDamg$, A_2 is a better solution, and with respect to $Delay$ and $Cost$, A_1 is a better solution. Note that $RiskPrb$ and $Portability$ are irrelevant. To reduce the number of swaps needed in the next step, $RiskDamg$ must be swapped with either $Delay$ or $Cost$, aiming to remove $RiskDamg$, so A_1 would dominate A_2 with respect to all relevant goals.

Rule 2: Pick the most reusable swaps. When stakeholders make a swap, their input can be reused for another alternative, without further consultation with human stakeholders. This reduces the number of swap queries from stakeholders. A swap from previous steps can be reused in the current step if the goals and their values are identical with the previous swap. Assume stakeholders have made a swap as $(g_x : x \rightarrow x' \Leftrightarrow g_y : y \rightarrow y')$ in a previous cycle. Now to decide between two alternatives such as A and B in a different cycle, this swap is reusable iff:

- Consequences of A on g_x and g_y are equal to x and y
- Consequences of B on g_x is x'

Under these conditions, alternative A can be replaced with A' where consequences of A' on g_x and g_y are revised to be x' and y'. Thus g_x becomes an irrelevant goal and can be removed.

Goal Two: Suggesting Easy Swaps: In addition to considering swaps reusability, the algorithm suggests swaps that decision stakeholders would be willing to make. Hammond et al. [14] suggest making the easiest swaps first, e.g., money is an easy goal to swap. What would make a swap easy for stakeholders? For example, stakeholders may easily agree to improve on a goal that is not sufficiently satisfied and compensate it with decreasing the satisfaction level of a requirement that is highly satisfied, to reach a balance among goals.

In addition, if consequences of two alternatives on the first goal of the swap are close, with an insignificant change, such a goal can become irrelevant. In this way, the goals that do not differentiate alternatives are eliminated from the problem earlier. The tool swaps the first goal with another goal for which consequences of alternatives are highly differentiable, because it would be more probable that the revised virtual alternative after applying the swap is still better than the other alternative.

Rule 3, make the easiest swap: When comparing two alternatives A and B, two goals such as g_1 and g_2 are swapped where the satisfaction level of A on g_1 is minimum compared to any other goal, and the satisfaction level of g_2 is the highest level compared to other goals. We define a distance factor between alternatives on every goal, which aggregates these desired properties into a value. The distance factor on the goal g_x is $\triangle(A, B, g_x)$ and calculated as:

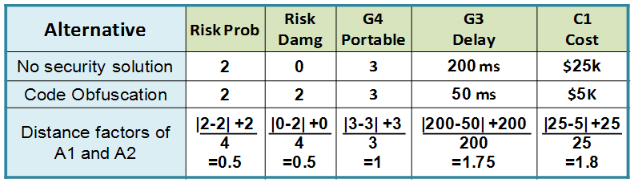
$$\triangle(A, B, g_x) = \frac{|a_x - b_x| + a_x}{max_{g_x}}$$

where a_x and b_x are the consequences of A and B on g_x. max_{g_x} is the maximum satisfaction level of g_x in the consequence table, and by dividing $|a_x - b_x| + a_x$ to the maximum value, distance factors of different alternatives are normalized to values in the scale of 0 to 2.

For example, the max of G_4 in Figure 1 is ✓ (fully satisfied). For the goals that need to minimized, the lower their value is, the higher the satisfaction level would be. Thus, if the consequence of A on g_x is a_x, then a_x is replaced with $max_{g_x} - a_x$. The tool revises the satisfaction level of goals in the consequence table in Figure 1 are as:

	$RiskProb$	$RiskDamg$	G_4	G_3	c_1
A_1	Medium Low	0	✓	200 ms	\$25 K
A_2	Medium Low	Medium Low	✓	50 ms	\$5 K
A_3	Medium Low	Medium High	✓	0 ms	\$10 K
A_4	Medium High	Medium High	✗.	150 ms	\$0 K
max_g	Medium High	Medium High	✓	200 ms	\$25 K

These modifications to the consequence table are only used for calculating the distance factor. Figure 2 shows the distance factors of A_1 and A_2 on the goals. (For calculations, the interval scale of Low to High is mapped to the interval values of 1 to 5, and ✓, ✓., ✗., and ✗ are mapped to 3, 2, ,1, 0.)



Alternative	Risk Prob	Risk Damg	G4 Portable	G3 Delay	C1 Cost
No security solution	2	0	3	200 ms	\$25k
Code Obfuscation	2	2	3	50 ms	\$5K
Distance factors of A1 and A2	|2-2| +2 4 =0.5	|0-2| +0 4 =0.5	|3-3| +3 3 =1	|200-50| +200 200 =1.75	|25-5| +25 25 =1.8

Fig. 2. Distance factors of alternatives A_1 and A_2 on DRM player goals

By applying rule 1, we concluded that $RiskDamg$ must be swapped with either $Delay$ or $Cost$. Based on rule 3, the lowest distance factor ($RiskDamg$) must be swapped with the highest distance factor ($Cost$).

Rule 4, Swap goals with tangible scales: Tangible goals that are measured in absolute values, such as costs or delays, are easier to trade [14]. The final rule for suggesting the next swap is selecting goals that are measured in more granular and tangible scales, because dealing with tangible factors is easier for stakeholders. For example, costs in terms of money is more tangible than the risk level expressed as Medium, Low, High. Therefore, the tool prefers goals that are measured in absolute values to goals measured by percentages, percentages are preferred to ordinal values, and ordinal values are preferred to qualitative labels.

4.2 The Automated Swaps Suggestion Tool

The automated Even Swaps tool takes a set of goals and a consequence table, and in each round of the Even Swaps method, identifies a list of potential goals to be swapped next. The list is first generated by applying rule 1. Then the list is trimmed by only keeping the most reusable swaps (rule 2). To find the best swap, the tool applies rule 3, and if still there are more than one possible swap for the next step, rule 4 is applied. A demo of the proposed tool in this paper is available at [17]. The core features of the tool and a graphical user interface is developed (in Java), and further improvements and tests are undergoing.

Given m goals, in each round, at most $m - 1$ swaps are made, and for n alternatives, at most $n - 1$ rounds of Even Swaps are needed. Thus, in the worst case scenario, with $m - 1 \times n - 1$ swaps the best solution is identified. By reusing swaps and applying rule 1, we aim to minimize this number.

5 Conclusions and Limitations

In this work, we adopt and enhance the Even Swaps [14] method for analyzing trade-offs among requirements when multiple alternative design solutions satisfy different requirements to some extent. The main contribution of this tool is applying a set of rules for suggesting next swaps to the decision stakeholders. The algorithm and prototype tool are able to handle different types of input data: absolute and ordinal values in different scales.

A threat to practicality of the tool is the diversity of evaluation scales in the consequence table. Stakeholders may not be able to swap a goal measured in absolute values with a goal that is evaluated by qualitative labels such as ✓ and ✗.

References

1. M. S. Feather, S. L. Cornford, K. A. Hicks, J. D. Kiper, and T. Menzies, "A broad, quantitative model for making early requirements decisions," *IEEE Software*, vol. 25, pp. 49–56, 2008.
2. *Security Risk Management Guide*, Microsoft Corporation: Microsoft Solutions for Security and Compliance and Microsoft Security Center of Excellence, 2006.
3. E. Yu, "Modeling Strategic Relationships for Process Reengineering," Ph.D. dissertation, University of Toronto, 1995.
4. P. Giorgini, J. Mylopoulos, E. Nicchiarelli, and R. Sebastiani, "Formal reasoning techniques for goal models," *Journal of Data Semantics*, vol. 1, pp. 1–20, 2003.
5. L. Chung, B. A. Nixon, E. Yu, and J. Mylopoulos, *Non-Functional Requirements in Software Engineering*. Kluwer Academic, 1999.
6. J. Horkoff and E. Yu, "A Qualitative, Interactive Evaluation Procedure for Goal- and Agent-Oriented Models," in *CAiSE Forum*, 2009.
7. P. Giorgini, J. Mylopoulos, and R. Sebastiani, "Goal-oriented requirements analysis and reasoning in the tropos methodology," *Eng. Appl. Artif. Intell.*, vol. 18, no. 2, pp. 159–171, 2005.
8. W. Ma, L. Liu, H. Xie, H. Zhang, and J. Yin, "Preference model driven services selection," in *Proc. of CAiSE'09*, 2009, pp. 216–230.
9. P. Giorgini, G. Manson, and H. Mouratidis, "On security requirements analysis for multi-agent systems." in *SELMAS'03*, 2003.
10. Y. Asnar and P. Giorgini, "Modelling risk and identifying countermeasure in organizations," 2006, pp. 55–66.
11. E. Letier and A. van Lamsweerde, "Reasoning about partial goal satisfaction for requirements and design engineering," in *SIGSOFT '04/FSE-12*, 2004, pp. 53–62.
12. H. P. In, D. Olson, and T. Rodgers, "Multi-criteria preference analysis for systematic requirements negotiation," in *Proc. of the COMPSAC'02*, ser. COMPSAC '02, 2002, pp. 887–892.
13. T. Saaty, *The Analytic Hierarchy Process, Planning, Piority Setting, Resource Allocation*. New york: McGraw-Hill, 1980.
14. J. S. Hammond, R. L. Keeney, and H. Raiffa, *Smart choices : a practical guide to making better life decisions*. Broadway Books, 2002.
15. J. Mustajoki and R. P. Hämäläinen, "Smart-swaps - a decision support system for multicriteria decision analysis with the even swaps method," *Decis. Support Syst.*, vol. 44, no. 1, pp. 313–325, 2007.
16. C. Collberg and J. Nagra, *Surreptitious Software: Obfuscation, Watermarking, and Tamperproofing for Software Protection*. Addison-Wesley Professional, 2009.
17. (2011) Decision analysis tool demo, release 1, dcs, univ. of toronto, available at http://www.cs.toronto.edu/~gelahi/Release1/Release1.html.

Flexab – Flexible Business Process Model Abstraction

Matthias Weidlich, Sergey Smirnov, Christian Wiggert, and Mathias Weske

Hasso Plattner Institute, Potsdam, Germany
{matthias.weidlich,sergey.smirnov,mathias.weske}@hpi.uni-potsdam.de,
christian.wiggert@student.hpi.uni-potsdam.de

Abstract. Process models are a widely established means to capture business processes. Large organizations maintain process model collections with hundreds of process models. Maintenance of these collections can be supported by business process model abstraction. Given a detailed model, an abstraction technique derives a coarse grained process model that preserves the essential process properties. In this paper, we introduce Flexab, a tool that realizes flexible process model abstraction. Arbitrary groups of activities may be selected for abstraction. Flexab is realized in a mashup environment, which allows for creating different abstracted versions of a process model and comparing them on a single screen.

Keywords: Process Model Abstraction, Model Synthesis.

1 Introduction

In the last decades, there has been a remarkable uptake of business process management (BPM). This trend emerged largely independent of any business domain or organizational background. Organizations that adopt BPM often manage the knowledge about their business processes by means of process models. These models define how business activities are performed in coordination to achieve a certain goal [16]. Large organizations maintain collections of hundreds of process models. The sheer number along with potential overlap of process models are challenges regarding the maintenance of such model collections.

Business process model abstraction (BPMA) emerged as a technique to support the management of large model collections. Given a very detailed model, it abstracts the process model by preserving essential process properties and leaving out insignificant details. In this way, maintenance of model collections can be centered around the most fine-grained model – more abstract models are generated by an abstraction approach.

In this paper, we present Flexab, a tool for flexible business process model abstraction. The tool is based on the abstraction approach introduced in [13]. In contrast to other work on process model abstraction, e.g., [3, 7, 9, 10], it does not impose structural restrictions when selecting activities that should be grouped into more coarse-grained ones. Instead, it is flexible in the sense that arbitrary

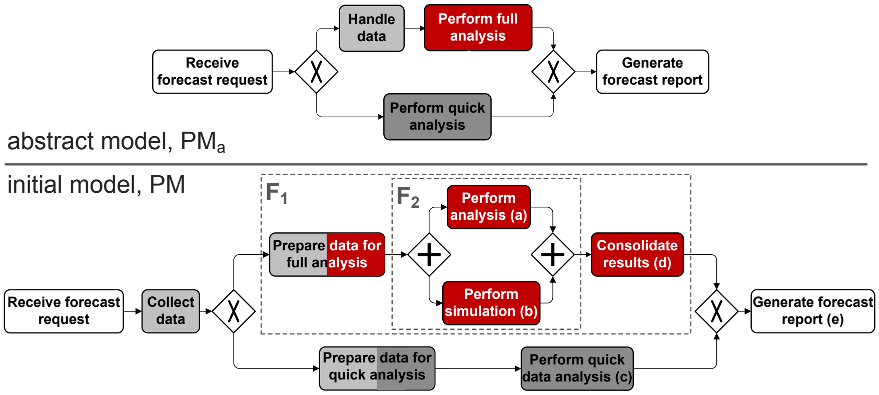


Fig. 1. Motivating example: initial model and activity grouping

groups of activities may be selected for abstraction. The question of how to define control flow dependencies for these arbitrary groupings in the abstracted model has been addressed in [13] using behavioral profiles. These profiles capture control flow relations between pairs of activities. Flexab implements the approach in a web-based environment. Using the Oryx framework [6], we created a mashup environment. This environment features gadgets for the visualization of process models and for providing the abstraction functionality. Using the Flexab gadgets, different abstracted versions of a common process model can be created and compared on a single screen. As such, our tool allows for different abstract views on a detailed process model at the same time.

The remainder of this paper is structured accordingly. The next section summarizes our approach to flexible abstraction of process models. Then, Section 3 introduces the implementation of this approach. We elaborate on the system architecture and explain the realization of all steps of the abstraction in detail. Finally, Section 4 reviews related work, before we conclude in Section 5.

2 Business Process Model Abstraction Approach

This section summarizes the approach to flexible abstraction of process models that was introduced in [13]. This approach focuses on the control flow perspective and has been defined for a generic graph model. The latter captures the commonalities of process modeling languages, i.e., a process model is a graph consisting of activities and control nodes that realize the routing behavior (aka gateways in BPMN and connectors in EPCs).

We illustrate the approach using the example depicted in Fig. 1. The lower model PM represents a detailed model of a forecasting process. This models contains several semantically related activities, indicated by the coloring in Fig. 1. These activities may be grouped to arrive at an abstract process model. Our abstraction approach allows for arbitrary grouping of activities, which may even be overlapping (indicated by a two-colored activity background in Fig. 1). This flexibility is not offered by existing approaches, which allow to aggregate only fragments, such as the groups $F1$ or $F2$ illustrated in Fig. 1.

Our abstraction approach comprises four steps. In the remainder of this section, we explain each of these steps.

1. derive the behavioral profile BP_{PM} of the process model PM
2. construct the behavioral profile BP_{PM_a} for the abstract process model PM_a
3. **if** a well-structured model with profile BP_{PM_a} exists
4. **then** create PM_a, **else** report to user.

1. Derivation of the Behavioral Profile $\boldsymbol{BP_{PM}}$. The approach leverages the notion of a behavioral profile. Such a profile captures behavioral characteristics of a process model by means of relations between pairs of activities. Two activities are said to be in strict order, if one occurs always before the other in every trace of the process model that contains both activities (e.g., (d) and (e) in Fig. 1). Activities that never occur together in a single trace are exclusive according to the behavioral profile (e.g., (c) and (d)). If two activities may occur in any order in a trace, then they are in interleaving order (e.g., (a) and (b)). For the class of process models considered by our approach (assuming the absence of behavioral anomalies such as deadlocks), the relations of the behavioral profile are computed in low polynomial time to the size of the model [15]. With BP_{PM}, we refer to the behavioral profile comprising the aforementioned relations for the model PM.

2. Construction of Behavioral Profile $\boldsymbol{BP_{PM_a}}$. As the next step, we require a user to select groups of activities in the detailed process model that should be aggregated in the abstracted model. For our example in Fig. 1, a user defines several aggregations for activities, such as the aggregation of activities *Prepare data for quick analysis* and *Perform quick data analysis* that yields an activity *Perform quick analysis.* Once aggregation dependencies have been defined, we leverage the behavioral profile BP_{PM} of PM to construct a behavioral profile BP_{PM_a} for the abstract model PM_a. This works as follows. For each pair of coarse-grained activities x, y in PM_a, we study the relations of the activities in PM that are aggregated into activities x and y. As a result, we obtain a dominating behavioral relation between the activities that are aggregated. This approach has the advantage that behavioral relations between activity pairs of PM_a are discovered independently of each other. For the setting in Fig. 1, for instance, we observe that both activities *Prepare data for quick analysis* and *Perform quick data analysis* are in strict order with *Generate forecast report.* Hence, the aggregated activity *Perform quick analysis* and activity *Generate forecast report* are in strict order in the behavioral profile BP_{PM_a}.

3. Behavioral Profile Well-Structuredness Validation. The creation of the behavioral profile for the abstract model may yield an inconsistent profile. That is, we may obtain a behavioral profile for which there does not exist a process model that satisfies certain requirements, e.g., that is free of behavioral anomalies and free of duplicated activities, and shows the relations of this profile. An example for an inconsistency would be a cyclic strict order dependency between activities

(x before y before z before x). The implementation in Flexab deviates from the synthesis proposed in [13], which is underspecified. Within Flexab, we analyze the behavioral profile BP_{PM_a} following an approach proposed for different behavioral relations to restructure process models [11]. Based on the profile relations, we create a graph that represents the different behavioral dependencies between activities. Then, a modular decomposition is applied to this graph. It identifies a hierarchy of modules, groups of activities that have equal dependencies with the remaining activities. The behavioral profile is well-structured if the decomposition yields a hierarchy of modules and none of them is unstructured. If the behavioral profile is well-structured, there exists a well-structured process model that is free of behavioral anomalies and shows the respective profile.

4. Abstract Model Synthesis from BP_{PM_a}. Given a well-structured behavioral profile for the abstract model, we create the abstract process model. All modules, groups of activities that have equal relations to all remaining activities, identified in the previous step directly translate into process model fragments. For instance, a module comprising activities that are all pairwise exclusive to each other is represented by an XOR-block containing the respective activities. Since the modular decomposition yields a hierarchy of modules, we are able to stepwise synthesis the process model. For our example, Fig. 1 illustrates the abstract model PM_a derived from the initial model PM. The model PM_a, for instance, reflects the strict order relation between activities *Perform quick analysis* and *Generate forecast report* derived before.

3 Process Model Abstraction using Flexab

In this section, we elaborate on Flexab—an application enabling process model abstraction. Flexab extends the Oryx framework, which we introduce first. Then, we describe the Flexab architecture and illustrate the usage to demonstrate the capabilities of Flexab.

Oryx. We implemented the business process model abstraction approach described in Section 2 within the Oryx Framework. Oryx is an extensible modeling framework bringing Web 2.0 technologies to business process designers. It allows for web-based modeling following a zero-installation approach. Oryx identifies each model by a URL, so that models can be shared by passing references rather than by exchanging model documents in email attachments. The framework can be extended in various directions. New languages are added by stencil sets that define explicit model element typing, rules of the composition and connection of elements, and the visualization of elements. Further, Oryx features a plugin infrastructure to add new functionality.

Oryx is organized into client and server components. The client component, the Oryx editor, realizes the modeling functionality. The editor is a JavaScript application running in a web-browser. The server component, the Oryx backend, stores process models, stencil sets, and fulfills other tasks, e.g., user management and rendering of various model representations (SVG, PNG, or PDF). The backend is implemented in Java.

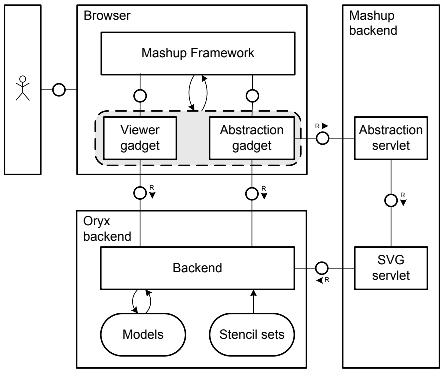
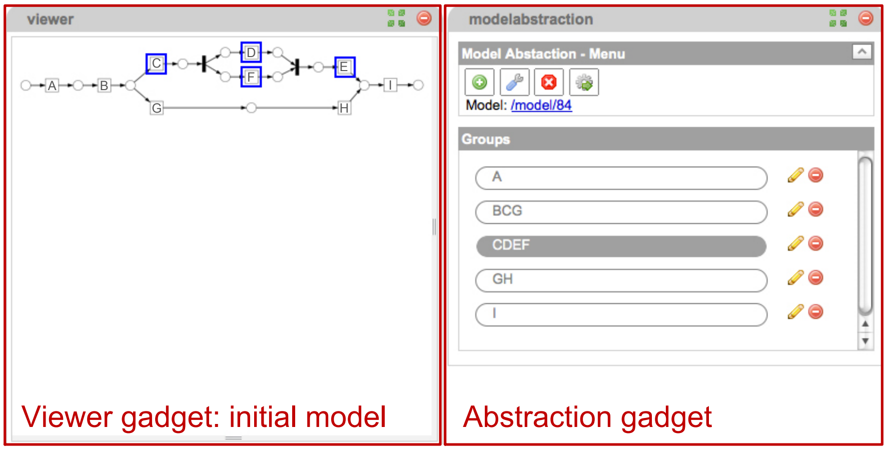


Fig. 2. Flexab architecture overview (FMC notation)

Oryx Mashup Framework. The Oryx editor addresses use cases that center around a single model, i.e., a designer edits one model at a time and does not need to trace dependencies with other models. However, several use cases, and process model abstraction is one of them, require the designer to observe several models simultaneously. The Oryx Mashup Framework provides an API for developing applications in which several models are manipulated on one screen. Similar to the Oryx Editor, the Mashup Framework is written in JavaScript and runs within a browser. The framework organizes functionality by *gadgets* and provides means to support communication between different gadgets. Each gadget not only accumulates business logic, but also has a UI representation. The UI components of gadgets are allocated on a dashboard. Typical gadgets provide model viewing functionality or enable selection of model elements. Hence, the Oryx Mashup Framework enables developers to create mashups for analyzing existing Oryx models and for concurrent interaction with several models.

FLEXAB. We have used the Oryx Mashup Framework as the basis for Flexab. Logically, the application is decomposed into the client-side and server-side components. The client-side component is built as an extension of the Oryx Mashup Framework. The server-side component is further distributed into the Oryx backend and Mashup backend, see Fig. 2. The communication between these three components is established by HTTP requests. The client-side component renders the user interface of the application. A viewer gadget presents the initial model that should be abstracted. The abstraction gadget, in turn, enables the user to define activity groups. This is supported by the viewer gadget to allow for populating groups with activities by simply selecting the activities in the viewer. Finally, another instance of a viewer gadget is used to show the abstract model.

Once the abstraction is triggered, the abstraction gadget sends the user-defined activity groups along with the initial process model to the abstraction servlet on the server side. Given this input, the abstraction servlet performs the abstraction algorithm and produces an abstract model. The abstraction servlet is supported by an SVG servlet that is responsible for the generation of a



Model: /model/84

Fig. 3. Screenshot of Flexab at the stage of activity group creation

SVG representation of the abstract model. To this end, it needs to retrieve the respective stencil set from the Oryx backend.

From a user perspective, abstracting a process model in Flexab works as follows. The user starts selecting the model to be abstracted. In response, the application caters two gadgets: a viewer gadget and an abstraction gadget, see Fig. 3. The user creates named activity groups, edits, and deletes the groups using the controls of the abstraction gadget. The viewer gadget not only renders the process model and provides zoom functionality, but also supports activity group creation: the user populates groups selecting activities directly in the model. Once the groups are finalized, the user initiates model transformation clicking the abstraction button in the abstraction gadget. Then, Flexab abstracts the model in the background and instantiates a new viewer gadget to visualize the result of abstraction. Fig. 4 presents the UI constellation in terms of the complete Mashup dashboard once model abstraction completes.

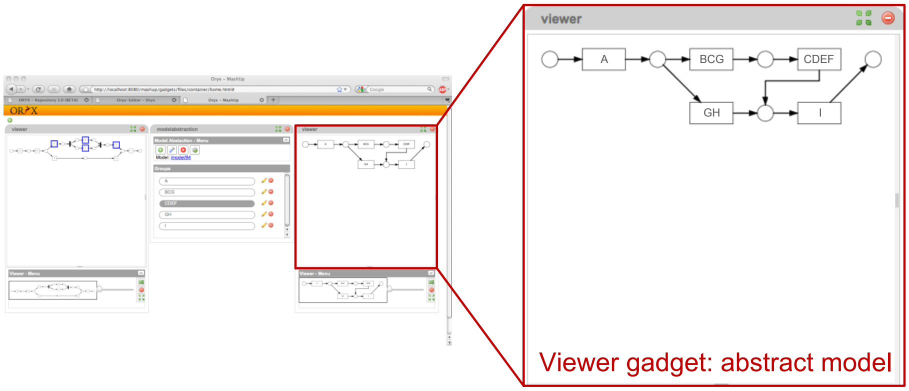


Fig. 4. Flexab presents the process model emerging from abstraction

4 Related Work

Flexab supports the user in the creation of an abstract process model given a detailed model. We identify three streams of related work, theoretical foundations of process model abstraction, applications implementing abstraction functionality, and research on process model generation.

In the recent years, a number of techniques for business process model abstraction emerged, e.g., [3, 4, 7–9, 12, 13]. All of these works investigate the theoretical principles of process model abstraction, while some address the implementation aspects as well. In [3, 4, 7–9, 13] the primary challenge of process model abstraction is addressed, i.e., structural model transformation. While the approaches of [3, 4, 9] build on an explicit definition of a fragment to aggregate, in [2, 10, 14] the fragments are discovered according to their properties. The abstraction approach implemented in this work equips the user with the most flexible way of activity aggregation. The question of how to identify model elements that are candidates for abstraction has been tackled in [7, 8, 12].

A few ideas on business process model abstraction found their way into implementations. The contribution of Bobrik, Reichert, and Bauer is realized in the Proviado system [3] and the approach presented in [9] has also been implemented in a prototype. As mentioned earlier, both approaches impose restrictions on the selection of the activities that are avoided by our approach. In the context of process mining, a mechanism for process simplification has been realized as a ProM plugin [8]. In contrast to our work, this simplification is guided by the occurrence frequency of activities in event logs. A system architecture for an application realizing model abstraction has been presented in [7].

The employed method for the synthesis of abstract process models from behavioral profiles belongs to the family of process model synthesis techniques. Most prominently, the alpha-algorithm constructs a process model given an event log [1]. The relations used in this algorithm differ to ours, since they are grounded on direct successorship of activities. A number of approaches based on Petri net formalism take the state space as an input for process model synthesis, e.g., [5].

5 Conclusion and Future Work

The theoretical aspects of business process model abstraction have been described in numerous papers. Up until now, however, very few implementations of these approaches have been presented. This paper showcases Flexab—an implementation of the business process model abstraction developed in [13]. Flexab builds on the Oryx framework. Hence, it brings together the functionality of process model abstraction and the Web 2.0 features of the Oryx framework including an extensible Mashup framework.

We have to reflect on some limitations of Flexab. The abstraction is currently restricted to Petri net models. Further, Flexab does not address the challenge of naming activities in abstract process models. In future work, we want to extend Flexab towards automation of process model abstraction. Since the current version of the tool requires the modeler to group model elements manually, the natural next step is to develop functionality for the automatic discovery of activity groups in process models.

Acknowledgments

The authors acknowledge the technical support of Gero Decker and Philipp Maschke from Signavio, a BPM company based in Berlin.

References

1. W. M. P. van der Aalst, A. J. M. M. Weijters, and L. Maruster. Workflow Mining: Discovering Process Models from Event Logs. *IEEE TKDE*, 16(9):1128–1142, 2004.
2. A. Basu and R.W. Blanning. Synthesis and Decomposition of Processes in Organizations. *ISR*, 14(4):337–355, 2003.
3. R. Bobrik, M. Reichert, and T. Bauer. View-Based Process Visualization. In *BPM 2007*, volume 4714 of *LNCS*, pages 88–95, Berlin, 2007. Springer.
4. J. Cardoso, J. Miller, A. Sheth, and J. Arnold. Modeling Quality of Service for Workflows and Web Service Processes. Technical report, University of Georgia, 2002.
5. J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Deriving Petri Nets from Finite Transition Systems. *IEEE TC*, 47(8):859–882, August 1998.
6. G. Decker, H. Overdick, and M. Weske. Oryx - Sharing Conceptual Models on the Web. In *ER*, volume 5231 of *LNCS*, pages 536–537. Springer, 2008.
7. R. Eshuis and P. Grefen. Constructing Customized Process Views. *DKE*, 64(2):419–438, 2008.
8. C. W. Günther and W. M. P. van der Aalst. Fuzzy Mining—Adaptive Process Simplification Based on Multi-perspective Metrics. In *BPM 2007*, volume 4714 of *LNCS*, pages 328–343, Berlin, 2007. Springer.
9. D. Liu and M. Shen. Workflow Modeling for Virtual Processes: an Order-preserving Process-view Approach. *ISJ*, 28(6):505–532, 2003.
10. A. Polyvyanyy, S. Smirnov, and M. Weske. The Triconnected Abstraction of Process Models. In *BPM 2009*, pages 229–244, Ulm, Germany, 2009. Springer.
11. A. Polyvyanyy, L. García-Bañuelos, and M. Dumas. Structuring acyclic process models. In *BPM 2010*, volume 6336 of *LNCS*, pages 276–293. Springer, 2010.
12. S. Smirnov, H. Reijers, Th. Nugteren, and M. Weske. Business Process Model Abstraction: Theory and Practice. Technical report, Hasso Plattner Institute, 2010. `http://bpt.hpi.uni-potsdam.de/pub/Public/SergeySmirnov/abstractionUseCases.pdf`.
13. S. Smirnov, M. Weidlich, and J. Mendling. Business Process Model Abstraction Based on Behavioral Profiles. In *ICSOC 2010*, volume 6470 of *LNCS*, pages 1–16, 2010.
14. A. Streit, B. Pham, and R. Brown. Visualization Support for Managing Large Business Process Specifications. *LNCS*, 3649:205–219, 2005.
15. M. Weidlich, J. Mendling, and M. Weske. Efficient Consistency Measurement based on Behavioural Profiles of Process Models. *IEEE TSE*, DOI: 10.1109/TSE.2010.96, 2010. In press.
16. M. Weske. *Business Process Management: Concepts, Languages, Architectures.* Springer, 2007.

A Tool for Automatic Enterprise Architecture Modeling

Markus Buschle, Hannes Holm, Teodor Sommestad, Mathias Ekstedt, and Khurram Shahzad

Industrial Information and Control Systems, KTH Royal Institute of Technology,
Osquldas v. 12, SE-10044 Stockholm, Sweden
{markusb, hannesh , teodors, mathiase, khurrams }@ics.kth.se,

Abstract. Enterprise architecture is an approach which aim to provide decision support based on organization-wide models. The creation of these models is however cumbersome as multiple aspects of an organization need to be considered. The Enterprise Architecture approach would be significantly less demanding if data used to create the models could be collected automatically.

This paper illustrates how a vulnerability scanner can be utilized for data collection in order to automatically create enterprise architecture models. We show how this approach can be realized by extending an earlier presented Enterprise Architecture tool. An example is provided through a case study applying the tool on a real network.

Keywords: Enterprise Architecture, Automatic data collection, Automatic instantiation, Software tool, Security Analysis

1 Introduction

Enterprise Architecture (EA) is a comprehensive approach for management and decision-making based on models of the organization and its information systems. An enterprise is typically described through dimensions such as Business, Application, Technology and Information. [7]. These pictographic descriptions are used for system-quality analysis to provide valuable support for IT and business decision-making [3].

As these models are intended to provide reliable decision support it is imperative that they capture all the aspects of an organization which are of relevance. Thus, they often grow very large and contain several thousands of entities and an even larger number of relationships in between them. The creation of such large models is both time and cost consuming, as lots of stakeholders are involved and many different pieces of information have to be gathered. During the creation process the EA models are also likely to become (partly) outdated [1]. Thus, in order to provide the best possible decision support it needs to be ensured that EA models both are holistic and reflect the organizations current state.

Automatic data collection and model creation would be preferable as this would mean a reduced modeling effort and an increased quality of the collected data. In current EA tools two approaches addressing automatic data collection can be found. The most common way is to import models that are made in 3rd party software. For example, BizzDesign Architect [2] can import from office applications. Thereby the automation aspect is the fact that data is reused and does not need to be manually entered if it is already available. The interpretation of data documented in the third-party software can however be resource- and time consuming, thus contradicting parts of the purpose with automatic data collection. Other tools such as for example Troux [11] allow the usage of SQL queries in order to load information from available data bases. This approach focuses on the extraction of the data-model and thereby the automatic creation of the information architecture as well as the business architecture based on process descriptions and similar documents.

In this paper we present how the Enterprise Architecture Analysis Tool [3] has been extended in order to automatically instantiate elements in EA models based on results from network scans. In comparison to the previously described approaches of other tools our implementation focuses on the Application and Technology layer of the organization. This information is gathered through an application of a vulnerability scanner that evaluates the network structure of an enterprise. Thereby attached network hosts and the functionality they provide can be discovered. Another difference is that the presented tool uses EA models for system-quality analysis, whereas commercial applications focus on modeling. As a running example we illustrate how a meta-model designed for cyber security analysis [9] can be (partly) automatically instantiated. The presented implementation is generic and can be used to support any kind of EA analysis.

The remainder of this paper is structured as follows. Section two describes the components used to realize the implementation and introduces into the meta-model that is used as running example. Section three describes how the information, which was automatically collected, is used to instantiate the meta-model for security evaluation. Section four exemplifies the tool application on real data collected by scanning a computer network used for security exercises. In section five the presented tool and the underlying approach are discusses as well as future work is described. Finally section six concludes the paper.

2 Preliminaries

This section describes the three components that we combined in order to automatically create EA models that can be used for security analysis. In subsection 2.1 the vulnerability scanner NeXpose [8], which is used for data collection, is explained. Subsection 2.2 describes the Enterprise Architecture Analysis Tool that is used to generate the models and evaluate them with regards to security aspects. Subsection 2.3 briefly introduces CySeMoL, the used meta-model which is partly instantiated using the automated data collection. The overall architecture can be seen in figure 1.

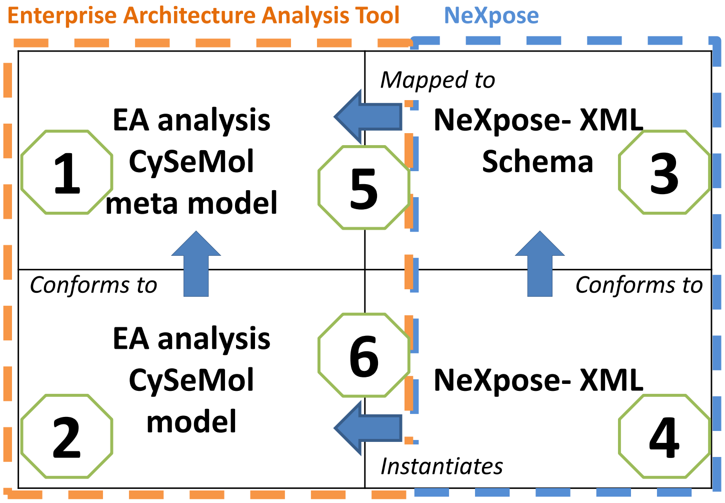

Enterprise Architecture Analysis Tool
NeXpose
Mapped to
EA analysis CySeMol meta model
1
5
NeXpose- XML Schema
3
Conforms to
Conforms to
EA analysis CySeMol model
6
NeXpose- XML
2
4
Instantiates

Fig. 1. The used architecture

2.1 NeXpose

The vulnerability scanner NeXpose was chosen in this project as it has demonstrated good results in previous tests [5].

NeXpose [8] is an active (i.e. it queries remote hosts for data) vulnerability scanner capable of both authenticated and unauthenticated scans. Authenticated scans involve providing the scanner with user accounts to hosts. They are typically less disturbing to normal operations and providing a higher degree of accuracy. However, it is not always the case that credentials are readily available for the individual(s) performing a scan.

NeXpose provides information regarding the network architecture in terms of all devices which are communicating over TCP or UDP, e.g. computers, firewalls and printers. The scanner identifies the operating systems or firmware that is running on the scanned devices and any services that are running. If the scanner is given credentials it is also able to assess all applications (and versions thereof) installed on a device and all user/administrator accounts on that device.

More security related functions of the scanner include that it can check for both software flaws and configuration errors. It is also capable of performing web application scans. NeXpose has approximately 53000 current signatures in its engine, with every signature corresponding to a certain vulnerability. NeXpose is also SCAP-compliant [6] and thus compliant with a suite of six commonly used protocols developed by the National Institute of Standards and Technology (NIST): i) Extensible Configuration Checklist Description Format (XCCDF), ii) Open Vulnerability and Assessment Language (OVAL), iii) Common Platform Enumeration (CPE), iv) Common Configuration Enumeration (CCE), v) Common Vulnerabilities and Exposures (CVE) and vi) Common Vulnerability Scoring System (CVSS).

2.2 Enterprise Architecture Analysis Tool

In [3] we presented a tool for EA analysis. This tool consists of two parts to be used in succession. The first component allows the definition of meta-models to describe a certain system quality of interest (**1** in Figure 1). This is done according to the PRM formalism [4] in terms of classes, attributes, and relations between them. Thereafter an execution of the second component is performed in order to describe an enterprise as an instantiated model (**2** in Figure 1), which is compliant to the previously defined meta-model. As the PRM formalism supports the expression of quantified theory the described enterprise can be evaluated with regards to the considered system quality described in the first component.

To use the results gained from NeXpose scans an extension of the tool was necessary. The result of NeXpose's scans can be exported to XML files (**4** in Figure 1), which are structured according to a schema definition file (XSD)[1] (**3** in Figure 1). We added the possibility to create mappings between XSD files and meta-models (**5** in Figure 1) in order to automatically instantiate the meta-model based on NeXpose's XML files (**6** in Figure 1). The used mapping is discussed in section 3.

2.3 CySeMoL

This paper exemplifies the mapping functionality by instantiating a subset of the meta-model of the CySeMoL (Cyber Security Modeling Language)[10]. This modeling language follows the abstract model presented in [9] and uses the PRM formalism to estimate the value of security attributes from an architecture model. Its meta-model covers both technical and organizational aspects of security and does in total contain 20 entities, 30 entity-relationships and a number of inter-dependent attributes. Four of these entities and three of its relationships can be mapped to elements produced by NeXpose. This subset of CySeMoL is depicted in the left part of Figure 2. While only a subset of the total number of entities and relations could be instantiated, this subset includes entities and relations which are of high multiplicity in enterprises, and thus require lots of effort to model.

3 The mapping

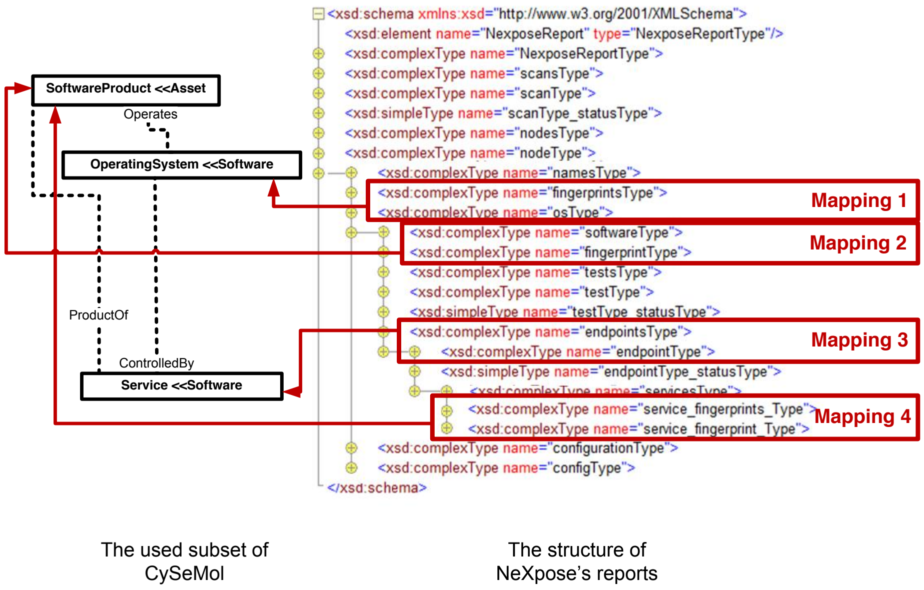
In this section we describe how we matched the structure of NeXpose's results to entities of the CySeMol language in order to instantiate the language based on scans. As described earlier, this was done based on the XSD file that describes the structure of the reports.

For our implementation we used four elements that a NeXpose result contains. At first we mapped *fingerprintsType* and *osType* to the *OperatingSystem*

[1] The XSD file (Report_XML_Export_Schema.xsd) is part of the NeXpose Community Edition that can be downloaded from http://www.rapid7.com

class of CySeMol, visualized as **Mapping 1** in figure 2. This allows us to determine the used operating system of a computer identified by NeXpose. The second mapping (**Mapping 2** in figure 2) relates *softwareType* and *fingerprintType* to *SoftwareProduct* in order to identify the software that is executed on the considered system. Thirdly (**Mapping 3**) we mapped *endpointsType* and *endpointType* to *Service* in order to identify at which ports services are provided by a machine. Finally a mapping between *service_fingerprints_ Type* and *service_fingerprint_ Type* to *SoftwareProduct* was made (**Mapping 4**) in order to describe the software that provide services on the machine of interest.

Additionally we considered the hierarchical structure of the XSD file in order to derive relationships. This made it possible to add the relationships *Operates*, *ControlledBy*, and *ProductOf* as they are shown in Figure 2.



```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="NexposeReport" type="NexposeReportType"/>
  <xsd:complexType name="NexposeReportType">
  <xsd:complexType name="scansType">
  <xsd:complexType name="scanType">
  <xsd:simpleType name="scanType_statusType">
  <xsd:complexType name="nodesType">
  <xsd:complexType name="nodeType">
      <xsd:complexType name="namesType">
      <xsd:complexType name="fingerprintsType">
      <xsd:complexType name="osType">
          <xsd:complexType name="softwareType">
          <xsd:complexType name="fingerprintType">
          <xsd:complexType name="testsType">
          <xsd:complexType name="testType">
          <xsd:simpleType name="testType_statusType">
          <xsd:complexType name="endpointsType">
              <xsd:complexType name="endpointType">
              <xsd:simpleType name="endpointType_statusType">
                  <xsd:complexType name="servicesType">
                  <xsd:complexType name="service_fingerprints_Type">
                  <xsd:complexType name="service_fingerprint_Type">
      <xsd:complexType name="configurationType">
      <xsd:complexType name="configType">
</xsd:schema>
```

Fig. 2. The implemented mapping

4 Example

In this section we describe how we tested the implementation on a real network. We give a brief introduction to the background of the collected data. Afterwards we depict how the resulting auto-generated model looks like.

4.1 The setup

The main experimental setup was designed by the Swedish Defence Research Agency (FOI) in Linköping, Sweden with the support of the Swedish National

Defence College (SNDC). Also, a group of computer security specialists and computer security researchers originating from various northern-European governments, military, private sectors and academic institutions were part of designing the network architecture.

The environment was set to describe a simplified critical information infrastructure at a small electrical power utility. The environment was composed of 20 physical PC servers running a total of 28 virtual machines, divided into four VLAN segments. Various operating systems and versions thereof were used in the network, e.g. Windows XP SP2, Debian 5.0 and Windows Server 2003 SP1. Each host had several different network services operating, e.g. web-, mail-, media-, remote connection- and file sharing services. Furthermore, every host was more or less vulnerable through software flaws and/or poor configurations.

4.2 The result

We performed a NeXpose scan on the setup environment and thereafter applied the mapping as presented in chapter 3. The resulting auto generated model consists of 28 instances of CySeMol's *OperatingSystem* class. Furthermore 225 instances of the *Service* class and 141 instantiations of the *SoftwareProduct* class were automatically generated. The generated components were related based on the relations that are specified in CySeMol. Figure 3 shows the resulting model exemplary for one computer of the environment as the full model is to big to be shown here.

5 Discussion and future work

This paper demonstrates that vulnerability scanners can provide useful support for the creation of EA models. As mentioned earlier, the results of a scan do not deliver a complete EA model, but require some completion work. The application of an automated scan however significantly reduces modeling effort and provides an EA analyst with a model stub which he or she can complement with other types of data.

The validity and reliability of the proposed approach can be discussed from two different viewpoints: i) how much of the meta-model that can be captured, both in scope (i.e. how much of the meta-model that can be instantiated) and context (i.e. if the scanner provides all the information needed to accurately capture the context of a variable), and ii) how accurate a vulnerability scanner is at assessing the instantiated variables. Regarding i), most of the more modeling intensive concepts of CySeMoL are captured and all context are accurate. That is, the scanner provides e.g. all the information regarding vulnerabilities that CySeMoL requires. Regarding ii), the scanning accuracy in terms of assessing vulnerabilities is studied in [5]. The accuracy in terms of assessing software, operating systems and such is something that will be examined in future works.

It would also be interesting to look at other variables provided by automated vulnerability scanning, e.g. user accounts of systems. Furthermore, automated

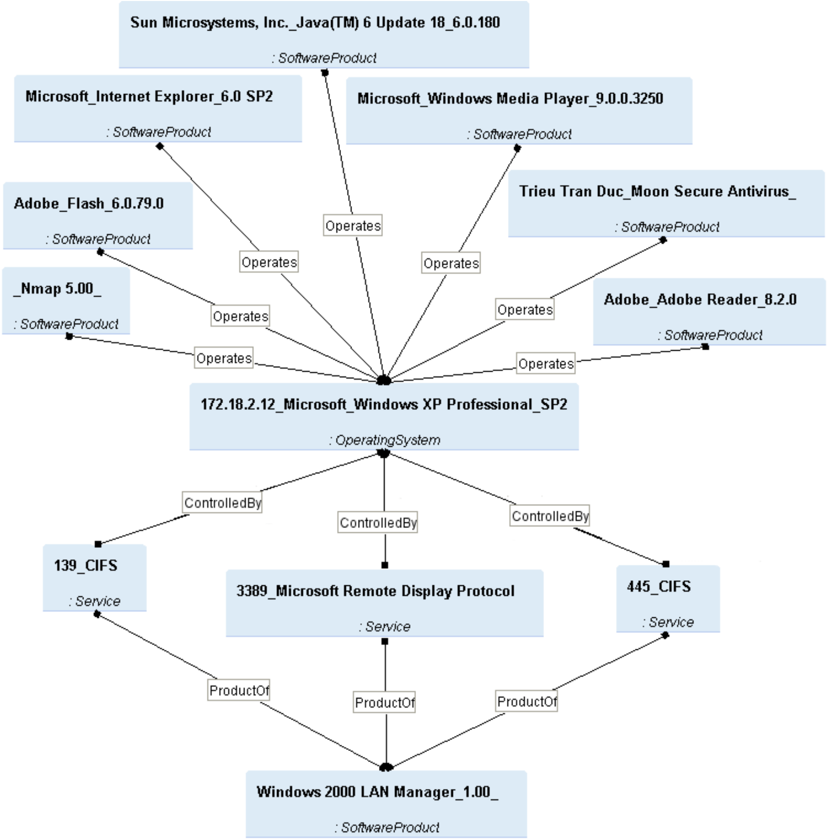


Fig. 3. The implemented mapping

scanning could be mapped to more commonly used EA frameworks such as ArchiMate [7] to increase the usage of the method.

Additionally in future work it might be investigated how other data sources can be used in order to provide input to automatic model creation and further reduce the manual tasks necessary. Examples of such sources are access control lists, ERP systems, accounting systems and UDDI registries. Especially how automatic data collection for the domains that so far not have been considered (the Business Layer and the information architecture) can be carried out, needs to be investigated. The long-term goal is to minimize the manual effort required to generate EA models.

The fact that enterprises are changing in the course of time is an important aspect too. The support for periodic scans leading to an automatic model update might therefore be implemented in the present tool as well.

It is also possible to collect information on vulnerabilities of services and software. This is something that we aim to incorporate in a future project in order to improve the analysis functionality.

6 Conclusion

In this paper we presented an extension of our previously developed tool that allows the automatic generation of elements for Enterprise Architecture models. The input for these models is provided by a vulnerability scanner, which was used to identify infrastructure elements and applications that were part of a computer network. Our implementation is generic even though CySeMol, a meta-model for security analysis, was used as a running example. The data gained from the vulnerability scanner can be used to instantiate any meta-model, as soon as a mapping has been defined. The scan with NeXpose took less than an hour and the creation of the EA model using that data was next to instantaneous. Thus, it should be a viable option for EA architects. We have also illustrated the architecture of our implementation and described used components in detail. Finally, we have presented a practical application based on real data of our implementation. Thereby we have shown the feasibility of our approach.

References

1. Aier, S., Buckl, S., Franke, U., Gleichauf, B., Johnson, P., Närman, P., Schweda, C., Ullberg, J.: A survival analysis of application life spans based on enterprise architecture models. In: 3rd International Workshop on Enterprise Modelling and Information Systems Architectures, Ulm, Germany. pp. 141–154 (2009)
2. BiZZdesign: BiZZdesign Architect. `http://www.bizzdesign.com` (Mar 2011)
3. Buschle, M., Ullberg, J., Franke, U., Lagerström, R., Sommestad, T.: A tool for enterprise architecture analysis using the prm formalism. In: CAiSE2010 Forum PostProceedings (2010)
4. Friedman, N., Getoor, L., Koller, D., Pfeffer, A.: Learning probabilistic relational models. In: Proc. of the 16th International Joint Conference on Artificial Intelligence. pp. 1300–1309. Morgan Kaufman (1999)
5. Holm, H., Sommestad, T., Almroth, J., Persson, M.: A quantitative evaluation of vulnerability scanning. Information Management & Computer Security (to be published)
6. Johnson, C., Quinn, S., Scarfone, K., Waltermire, D.: The technical specification for the security content automation protocol (SCAP). NIST Special Publication 800, 126 (2009)
7. Lankhorst, M.M.: Enterprise Architecture at Work: Modelling, Communication and Analysis. Springer, Berlin, Heidelberg, Germany, 2^{nd} edn. (2009)
8. Rapid7: NeXpose. `http://www.rapid7.com/` (Mar 2011)
9. Sommestad, T., Ekstedt, M., Johnson, P.: A probabilistic relational model for security risk analysis. Computers & Security 29(6), 659–679 (2010)
10. Sommestad, T., Ekstedt, M., Nordström, L.: A case study applying the Cyber Security Modeling Language (2010)
11. Troux Technologies: Metis. `http://www.troux.com/products/` (Mar 2011)

Discovering Hierarchical Process Models Using ProM

R.P. Jagadeesh Chandra Bose[1,2], Eric H.M.W. Verbeek[1] and Wil M.P. van der Aalst[1]

[1] Department of Mathematics and Computer Science, University of Technology, Eindhoven, The Netherlands
[2] Philips Healthcare, Veenpluis 5–6, Best, The Netherlands
{j.c.b.rantham.prabhakara,h.m.w.verbeek,w.m.p.v.d.aalst}@tue.nl

Abstract. Process models can be seen as "maps" describing the operational processes of organizations. Traditional process discovery algorithms have problems dealing with fine-grained event logs and less-structured processes. The discovered models (i.e., "maps") are spaghetti-like and are difficult to comprehend or even misleading. One of the reasons for this can be attributed to the fact that the discovered models are flat (without any hierarchy). In this paper, we demonstrate the discovery of hierarchical process models using a set of interrelated plugins implemented in ProM.[3] The hierarchy is enabled through the automated discovery of abstractions (of activities) with domain significance.

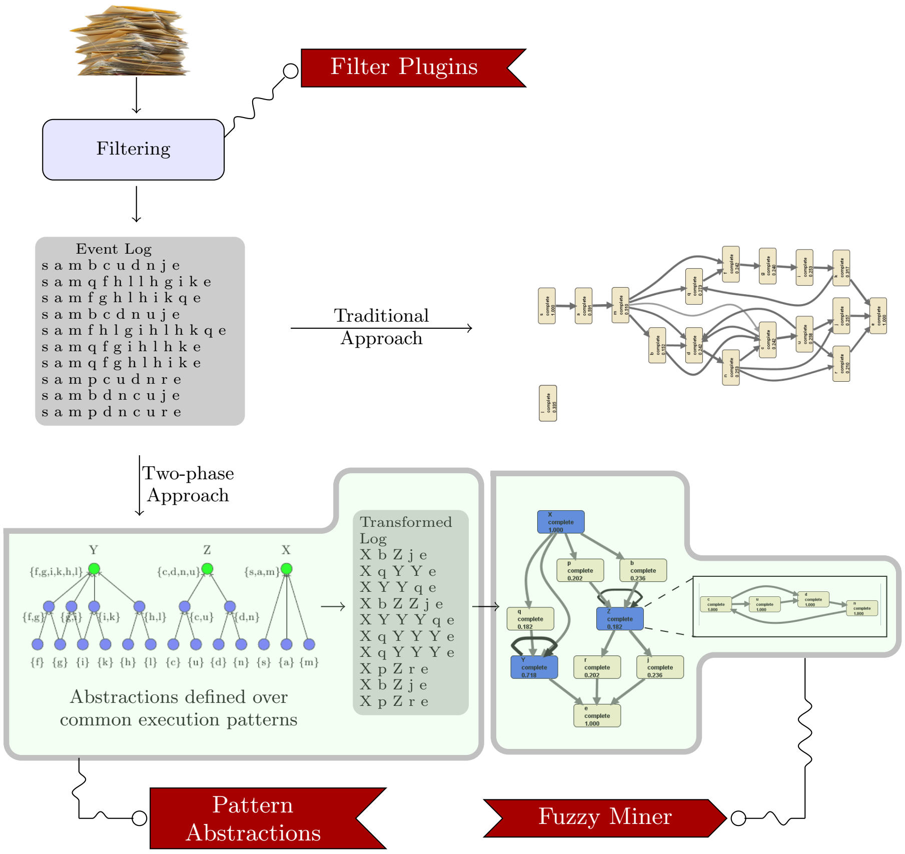
Keywords: process discovery, process maps, hierarchical models, abstractions, common execution patterns

1 Introduction

We have applied process mining techniques in over 100 organizations. These practical experiences revealed two problems: (a) processes tend to be less structured than what stakeholders expect, and (b) events logs contain fine-grained events whereas stakeholders would like to view processes at a more coarse-grained level. In [1], we showed that common execution patterns (e.g., tandem arrays, maximal repeats etc.) manifested in an event log can be used to create powerful *abstractions*. These abstractions are used in our *two-phase approach to process discovery* [2]. The first phase comprises of pre-processing the event log based on abstractions (bringing the log to the desired level of granularity) and the second phase deals with discovering the process maps while providing a seamless zoom-in/out facility. Figure 1 highlights the difference between the traditional approach to process discovery and our two-phase approach. Note that the process model (map) discovered using the two-phase approach is much simpler.

The *two-phase approach to process discovery* [2] enables the discovery of hierarchical process models. In this paper, we demonstrate the discovery of hierarchical process models using a chain of plugins implemented in ProM. The chain of plugins and their order of application is illustrated in Figure 2.

[3] ProM is an extensible framework that provides a comprehensive set of tools/plugins for the discovery and analysis of process models from event logs. See http://www.processmining.org for more information and to download ProM.



Event Log
s a m b c u d n j e
s a m q f h l l h g i k e
s a m f g h l h i k q e
s a m b c d n u j e
s a m f h l g i h l h k q e
s a m q f g i h l h k e
s a m q f g h l h i k e
s a m p c u d n r e
s a m b d n c u j e
s a m p d n c u r e

Transformed Log
X b Z j e
X q Y Y e
X Y Y q e
X b Z Z j e
X Y Y Y q e
X q Y Y Y e
X q Y Y Y e
X p Z r e
X b Z j e
X p Z r e

Fig. 1: Traditional approach versus our two-phase approach to process discovery. ProM plugins are used to filter the event log. ProM's *Pattern Abstractions* plugin and the *Fuzzy Miner* plugin are used to realize simple and intuitive models.

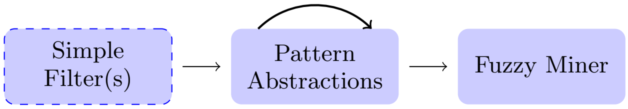


Fig. 2: The chaining of plugins that enables the discovery of hierarchical process models.

The event log may first be cleansed using some simple filters (e.g., adding artificial start/end events, filtering events of a particular transaction type such as considering only 'complete' events etc.). The *Pattern Abstractions* plugin is then applied on this filtered log one or several times. The Pattern Abstractions plugin has been implemented as a *log visualizer* in ProM and caters to *the discovery of common execution patterns, the definition of abstractions over them, and the pre-processing of the event log with these abstractions*. The transformed log (pre-processed log with abstractions) obtained in iteration i is used as the input for the Pattern Abstractions plugin in iteration $i+1$. It is this repetitive application of the Pattern Abstractions plugin that enables the definition of multiple levels of hierarchy (new abstractions can be defined over existing abstractions). During the pre-processing phase, for each defined abstraction, the Pattern Abstractions plugin generates a sub-log that captures the manifestation of execution patterns defined by that abstraction as its process instances. The Fuzzy Miner plugin [3] is then applied on the transformed log obtained after the last iteration. The Fuzzy Miner plugin in ProM has been enhanced to utilize the availability of sub-logs for the defined abstractions. Process models are discovered for each of the sub-logs and are displayed upon zooming in on its corresponding abstract activity.

Running Example. We use the workflow of a simple digital photo copier as our running example. The copier supports photocopying, scanning and printing of documents in both color and gray modes. The scanned documents can be sent to the user via email or FTP. Upon receipt of a job, the copier first generates an image of the document and subsequently processes the image to enhance its quality. Depending on whether the job request is for a copy/scan or print, separate procedures are followed to generate an image. For print requests, the document is first interpreted and then a rasterization procedure is followed to form an image. The image is then written on the drum, developed, and fused on to the paper.

We have modeled this workflow of the copier in CPN tools [4] and generated event logs by simulation. We use one such event log in this paper. The event log consists of 100 process instances, 76 event classes and $40,995$ events. The event log contains fine-grained events pertaining to different procedures (e.g., image processing, image generation etc.) mentioned above. An analyst may not be interested in such low level details. We demonstrate the discovery of the workflow at various levels of abstractions for this event log.

2 Pattern Abstractions Plugin

The basic building blocks of the Pattern Abstractions plugin are shown in Figure 3. Figures 4 and 5 illustrate these building blocks.

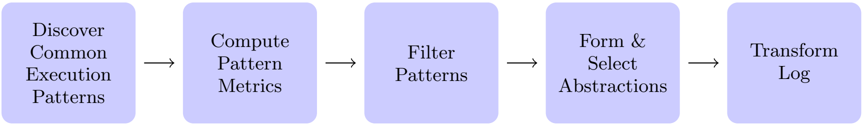
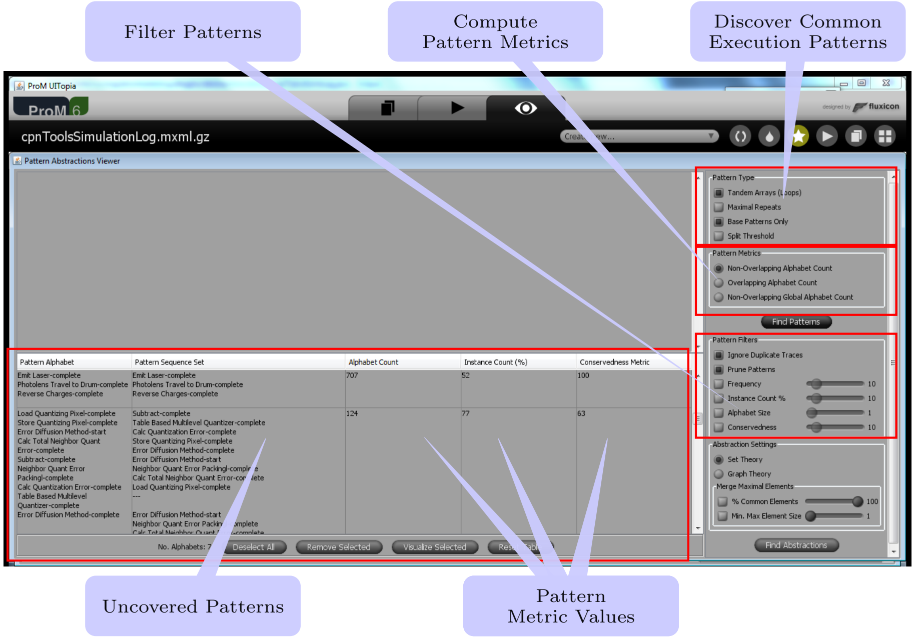

Discover Common Execution Patterns
Compute Pattern Metrics
Filter Patterns
Form & Select Abstractions
Transform Log

Fig. 3: Building blocks of the Pattern Abstractions plugin

- *Discover Common Execution Patterns:* The Pattern Abstractions plugin supports the discovery of tandem arrays (loop patterns) and maximal repeats (common subsequence of activities within a process instance or across process instances) [1]. These can be uncovered in *linear* time and space with respect to the length of the traces.



Pattern Type
- ☒ Tandem Arrays (Loops)
- □ Maximal Repeats
- ☒ Base Patterns Only
- □ Split Threshold

Pattern Metrics
- ● Non-Overlapping Alphabet Count
- ○ Overlapping Alphabet Count
- ○ Non-Overlapping Global Alphabet Count

Find Patterns

Pattern Filters
- ☒ Ignore Duplicate Traces
- ☒ Prune Patterns
- □ Frequency 10
- □ Instance Count % 10
- □ Alphabet Size 1
- □ Conservedness 10

Abstraction Settings
- ● Set Theory
- ○ Graph Theory

Merge Maximal Elements
- □ % Common Elements 100
- □ Min. Max Element Size 1

Find Abstractions

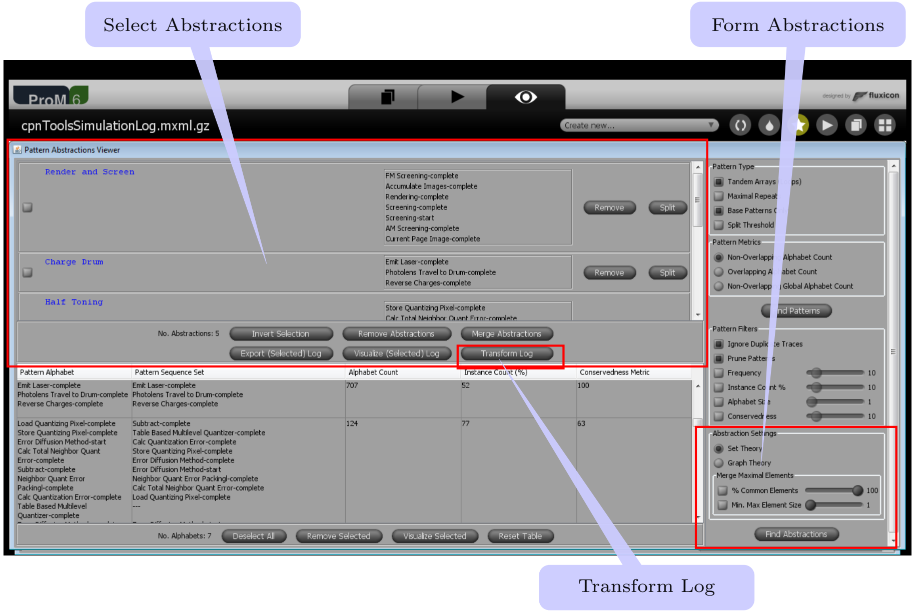
Pattern Alphabet	Pattern Sequence Set	Alphabet Count	Instance Count (%)	Conservedness Metric
Emit Laser-complete Photolens Travel to Drum-complete Reverse Charges-complete	Emit Laser-complete Photolens Travel to Drum-complete Reverse Charges-complete	707	52	100
Load Quantizing Pixel-complete Store Quantizing Pixel-complete Error Diffusion Method-start Calc Total Neighbor Quant Error-complete Subtract-complete Neighbor Quant Error Packingl-complete Calc Quantization Error-complete Table Based Multilevel Quantizer-complete Error Diffusion Method-complete	Subtract-complete Table Based Multilevel Quantizer-complete Calc Quantization Error-complete Store Quantizing Pixel-complete Error Diffusion Method-complete Error Diffusion Method-start Neighbor Quant Error Packingl-complete Calc Total Neighbor Quant Error-complete Load Quantizing Pixel-complete --- Error Diffusion Method-start Neighbor Quant Error Packingl-complete Calc Total Neighbor Quant Error-complete	124	77	63

No. Alphabets: 7 Deselect All Remove Selected Visualize Selected Reset Table

Fig. 4: The discovery of common execution patterns, computation of pattern metrics, filtering and inspection of patterns in the Pattern Abstractions plugin.

- *Compute Pattern Metrics:* Various metrics (e.g, overlapping and non-overlapping frequency counts, instance count etc.) to assess the significance of the uncovered patterns are supported.
- *Filter Patterns:* It could be the case that too many patterns are uncovered from the event log. To manage this, features to filter patterns that are less significant are supported.
- *Form and Select Abstractions:* Abstractions are defined over the filtered patterns. Patterns that are closely related are grouped together to form abstractions. The approach for forming abstractions is presented in [1]. Furthermore, various features to edit/select abstractions such as merging two or more abstractions and deleting activities related to a particular abstraction are supported. Figure 5 depicts a few abstractions defined over loop patterns for the copier event log e.g., *half-toning*, a procedure for enhancing the image quality, is uncovered as an abstraction.
- *Transform Log:* The event log is pre-processed by replacing activity subsequences corresponding to abstractions. A replaced activity subsequence is captured as a process instance in the sub-log for the corresponding abstract activity.

At any iteration, if n abstractions are selected, the Pattern Abstractions plugin generates a transformed log, and n sub-logs (one for each of the n chosen abstractions). We recommend to process for loop patterns in the initial iterations and maximal repeats in the subsequent iterations. For the example event log, we have performed three iterations. The transformed log after the third iteration has 19 event classes and 1601 events. In the process, we have defined various abstractions such as *half-toning, image processing, capture image*, etc.



cpnToolsSimulationLog.mxml.gz

Pattern Abstractions Viewer

□ Render and Screen
FM Screening-complete
Accumulate Images-complete
Rendering-complete
Screening-complete
Screening-start
AM Screening-complete
Current Page Image-complete

□ Charge Drum
Emit Laser-complete
Photolens Travel to Drum-complete
Reverse Charges-complete

Half Toning
Store Quantizing Pixel-complete
Calc Total Neighbor Quant Error-complete

No. Abstractions: 5

Pattern Alphabet	Pattern Sequence Set	Alphabet Count	Instance Count (%)	Conservedness Metric
Emit Laser-complete Photolens Travel to Drum-complete Reverse Charges-complete	Emit Laser-complete Photolens Travel to Drum-complete Reverse Charges-complete	707	52	100
Load Quantizing Pixel-complete Store Quantizing Pixel-complete Error Diffusion Method-start Calc Total Neighbor Quant Error-complete Subtract-complete Neighbor Quant Error Packingl-complete Calc Quantization Error-complete Table Based Multilevel Quantizer-complete	Subtract-complete Table Based Multilevel Quantizer-complete Calc Quantization Error-complete Store Quantizing Pixel-complete Error Diffusion Method-complete Error Diffusion Method-start Neighbor Quant Error Packingl-complete Calc Total Neighbor Quant Error-complete Load Quantizing Pixel-complete ---	124	77	63

No. Alphabets: 7

Pattern Type
☒ Tandem Arrays (Loops)
□ Maximal Repeats
☒ Base Patterns Only
□ Split Threshold

Pattern Metrics
● Non-Overlapping Alphabet Count
○ Overlapping Alphabet Count
○ Non-Overlapping Global Alphabet Count

Pattern Filters
☒ Ignore Duplicate Traces
☒ Prune Patterns
□ Frequency 10
□ Instance Count % 10
□ Alphabet Size 1
□ Conservedness 10

Abstraction Settings
● Set Theory
○ Graph Theory

Merge Maximal Elements
□ % Common Elements 100
□ Min. Max Element Size 1

Fig. 5: The generation and selection of abstractions in the Pattern Abstractions plugin.

The Pattern Abstractions plugin supports additional features such as visualizing patterns and exporting the traces that contain the patterns.

3 (Enhanced) Fuzzy Miner Plugin

The Fuzzy Miner [3, 5] is a process miner that mines an event log for a family of process models using a "map" metaphor. As many maps exist that show the city of Amsterdam at different levels of abstraction, also different maps exist for a process model mined from an event log. In this map metaphor, an object of interest in Amsterdam (like the Rijksmuseum or the Anne Frank House) corresponds to a node in the process model, where streets (like the Kalverstraat or the PC Hooftstraat) correspond to edges in the model. For sake of convenience, we call a single map a *fuzzy instance* whereas we call a family of maps (like all Amsterdam maps) a *fuzzy model*.

Like high-level maps only show major objects of interest and major streets, high-level fuzzy instances also show only major elements (nodes and edges). For this purpose, the Fuzzy Miner computes from the log a significance weight for every element and an additional correlation weight for every edge. The higher these weights are, the more major the element is considered to be. Furthermore, the Fuzzy Miner uses a number of thresholds: Only elements that meet these thresholds are shown. As such, these thresholds correspond to the required level of abstraction: The higher these thresholds are, the higher the level of abstraction is. For sake of completeness we mention here that a fuzzy instance may contain clusters of minor nodes: If some objects of interest on the Amsterdam map are too minor to be shown by themselves on some map, they may be shown as a single (and major) object provided that they are close enough. For this reason, the Fuzzy Miner first attempts to cluster minor nodes into major cluster nodes, and only if that does not work it will remove the minor node from the map.

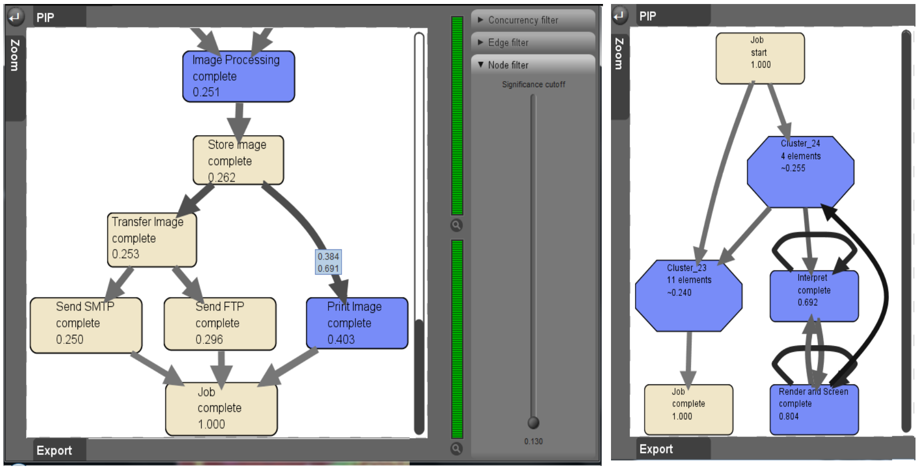


Fig. 6: Fuzzy model and instance

Figure 6 shows an example fuzzy model (left-hand side) and fuzzy instance (right-hand side). Note that both views show a fuzzy instance, but the fuzzy model view allows the user to change the thresholds (by changing the sliders) whereas the fuzzy instance view does not. The significance of a node is displayed as part of its label (for example, the node "Transfer Image" has a significance of 0.253), the significance of an edge is visualized using its wideness (the wider the edge, the more significant it is), and the correlation of an edge is visualized using its color contrast (the darker the edge is, the more correlated its input node and its output node are). The octagonal shaped nodes in the right-hand side view correspond to the cluster nodes (one of the cluster nodes contain 4 activities and the other contains 11 activities). All activities on the left hand side except "Job Complete" are contained in a cluster node on the right. Apparently, the significance weights for these nodes (0.262, 0.253, 0.250, 0.296 and 0.403) were too low to be shown, which indicates that the corresponding threshold was set to at least 0.403. Furthermore, the node "Interpret" (on the right) is highly self-correlated, whereas the nodes "Transfer Image" and "Send SMTP" (on the left) are moderately correlated.

The Fuzzy Miner has been enhanced to utilize the availability of sub-logs obtained from the Pattern Abstractions plugin for the chosen abstractions. Fuzzy models are discovered for each of the sub-logs and are displayed upon zooming in on its corresponding abstract activity. Abstract activities are differentiated from *other activities by means of a distinct color (a darker shade of blue, see also Figure 6).*

Figure 7 depicts the top-level process model of the copier example. This model is generated from the transformed log obtained after the third iteration of Pattern Abstractions plugin. The upper branch of the process model corresponds to the creation of the document image for print requests while the lower branch corresponds to image creation for copy/scan requests. The two branches meet after the image is formed and the image is subjected to some image processing functionality. The document is then printed or sent to the user via email or FTP. The lower level details of image creation, image processing, print image have been abstracted in this model. *The Pattern Abstractions plugin enables the discovery of such abstractions with strong domain (functional) significance.* Upon zooming in on the *Image Processing* abstraction, the process model depicted in Figure 8 is shown. This sub-process in turn contains another abstract activity viz., *Half Toning* (the level of hierarchy is two). Zooming in on this abstract activity displays the sub-process defining it as depicted in Figure 8. Figure 9 depicts two other abstractions.

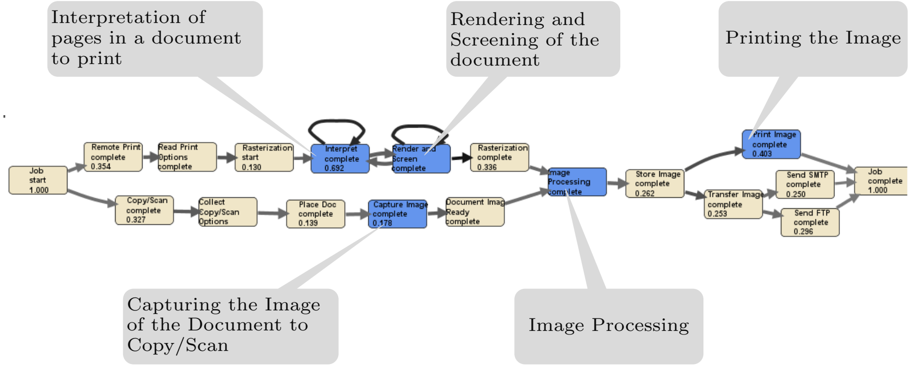

Interpretation of pages in a document to print
Rendering and Screening of the document
Printing the Image
Job start 1.000
Remote Print complete 0.354
Read Print Options complete
Rasterization start 0.130
Interpret complete 0.692
Render and Screen complete
Rasterization complete 0.336
Copy/Scan complete 0.327
Collect Copy/Scan Options
Place Doc complete 0.139
Capture Image complete 0.178
Document Imag Ready complete
Image Processing complete
Store Image complete 0.262
Print Image complete 0.403
Transfer Image complete 0.253
Send SMTP complete 0.250
Send FTP complete 0.296
Job complete 1.000
Capturing the Image of the Document to Copy/Scan
Image Processing

Fig. 7: The top level process model of the copier event log. Blue (dark colored) nodes are abstract activities that can be zoomed in. Upon zooming in, the sub-process defining the abstraction is shown.

In this fashion, using the chain of plugins presented in this paper, one can discover hierarchical process models.

4 Conclusions

We demonstrated the discovery of hierarchical process models using a chain of plugins implemented in ProM. The repetitive application of Pattern Abstractions plugin enables the discovery of multiple levels of hierarchy. We can use this approach to create maps that (i) depict desired traits, (ii) eliminate irrelevant details, (iii) reduce complexity, and (iv) improve comprehensibility.

Acknowledgments R.P.J.C. Bose and W.M.P. van der Aalst are grateful to Philips Healthcare for funding the research in process mining.

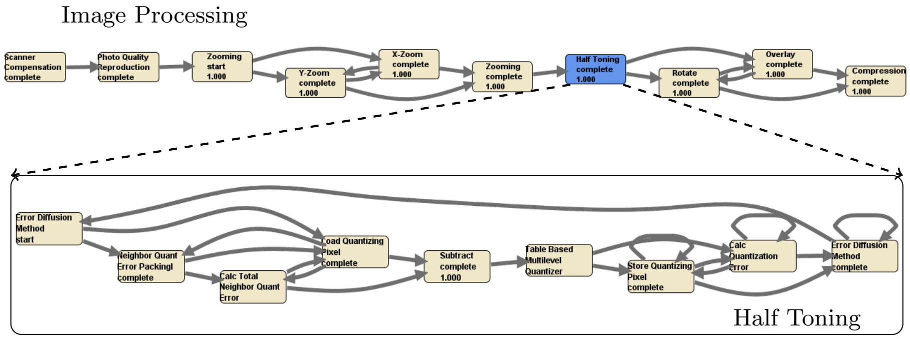


Image Processing
Scanner Compensation complete
Photo Quality Reproduction complete
Zooming start 1.000
Y-Zoom complete 1.000
X-Zoom complete 1.000
Zooming complete 1.000
Half Toning complete 1.000
Rotate complete 1.000
Overlay complete 1.000
Compression complete 1.000
Error Diffusion Method start
Neighbor Quant Error Packingl complete
Calc Total Neighbor Quant Error
Load Quantizing Pixel complete
Subtract complete 1.000
Table Based Multilevel Quantizer
Store Quantizing Pixel complete
Calc Quantization Error
Error Diffusion Method complete
Half Toning

Fig. 8: The sub-process captured for the abstraction 'Image Processing' (in the top-level model). This sub-process in turn contains another abstraction viz., 'Half Toning'. Upon zooming in on 'Half Toning', the sub-process defining that is shown.

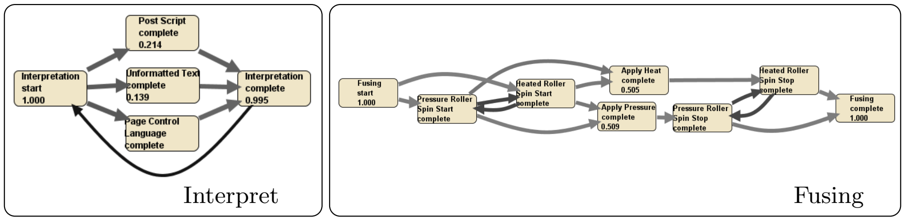

Interpretation start 1.000
Post Script complete 0.214
Unformatted Text complete 0.139
Page Control Language complete
Interpretation complete 0.995
Interpret
Fusing start 1.000
Pressure Roller Spin Start complete
Heated Roller Spin Start complete
Apply Heat complete 0.505
Apply Pressure complete 0.509
Pressure Roller Spin Stop complete
Heated Roller Spin Stop complete
Fusing complete 1.000
Fusing

Fig. 9: The sub-processes for the abstractions 'Interpret' and 'Fusing'. 'Interpret' is an abstract activity at the top-level of the process model while 'Fusing' is an abstract activity underneath the 'Print Image' abstraction.

References

1. Bose, R.P.J.C., van der Aalst, W.M.P.: Abstractions in Process Mining: A Taxonomy of Patterns. In Dayal, U., Eder, J., Koehler, J., Reijers, H., eds.: Business Process Management. Volume 5701 of LNCS., Springer-Verlag (2009) 159–175
2. Li, J., Bose, R.P.J.C., van der Aalst, W.M.P.: Mining Context-Dependent and Interactive Business Process Maps using Execution Patterns. In zur Muehlen, M., Su, J., eds.: BPM 2010 Workshops. Volume 66 of LNBIP., Springer-Verlag (2011) 109–121
3. Günther, C., van der Aalst, W.M.P.: Fuzzy Mining: Adaptive Process Simplification Based on Multi-perspective Metrics. In: International Conference on Business Process Management (BPM 2007). Volume 4714 of LNCS., Springer-Verlag (2007) 328–343
4. Vinter Ratzer, A., Wells, L., Lassen, H.M., Laursen, M., Qvortrup, J.F., Stissing, M.S., Westergaard, M., Christensen, S., Jensen, K.: CPN Tools for Editing, Simulating, and Analysing Coloured Petri Nets. In: 24th International Conference on Applications and Theory of Petri Nets (ICATPN). Volume 2679 of LNCS., Springer (2003) 450–462
5. Xia, J.: Automatic Determination of Graph Simplification Parameter Values for Fuzzy Miner. Master's thesis, Eindhoven University of Technology (2010)

Tool Support for Enforcing Security Policies on Databases

Jenny Abramov[2], Omer Anson[2], Arnon Sturm[1,] Peretz Shoval[1]

[1] Department of Information Systems Engineering
[2] Deutsche Telekom Laboratories (T-Labs)
Ben-Gurion University of the Negev
Beer Sheva 84105, Israel
jennyab@bgu.ac.il, oaanson@gmail.com,
sturm@bgu.ac.il, shoval@bgu.ac.il

Abstract. Security in general and database protection from unauthorized access in particular, are crucial for organizations. It has long been accepted that security requirements should be considered from the early stages of the development. However, such requirements tend to be neglected or dealt-with only at the end of the development process. The Security Modeling Tool presented in this study aims at enforcing developers, in particular database designers, to deal with database authorization requirements from the early stages of development. This software demonstration shows how the Security Modeling Tool assists to define organizational security policies and use them during the application development to create a secured database schema.

Keywords: Secure software engineering, database design, authorization.

1 Introduction

Data is the most valuable asset for an organization as its survival depends on the correct management, security, and confidentiality of the data [1]. In order to protect the data, organizations must secure data processing, transmission and storage. Developers of data-oriented systems always face problems related to security. This is the case as security and other non-functional requirements are usually ignored in the early stages of the development process.

To overcome these lacks, we provide a methodology that guides developers in the incorporation of particular organizational security policies, as well as verifying their correct application. In addition, the methodology enables the developer to transform the result into code.

The methodology incorporates ideas from two areas of expertise: in the area of *methodologies for system development*, we adopt the principle of integrating data and functional modeling at the early stages of the development, suggested by the Functional Object-Oriented Methodology (FOOM) [6]. Additionally, in the area of *domain engineering*, we adopt the principles suggested by the Application Based Domain Modeling (ADOM) approach [4]. ADOM supports building reusable assets on the one hand, and representing and managing knowledge in specific domains on the other hand. This knowledge guides the development of various applications in that domain and serves as a verification template for their correctness and completeness.

The developed methodology is supported by the Security Modeling Tool (SMT), which was tailored for its needs and was equipped with the required facilities. SMT enables the modeling of security patterns and enforces their correct use during application development. The knowledge captured in the security patterns is used to automatically verify that the application models are indeed secure, according to the defined patterns. Having a verified model, secure database schemata can be automatically generated.

The SMT is an Eclipse plug-in. The SMT uses, internally, libraries provided by other Eclipse plug-ins to complete many tasks. For instance, Eclipse Modeling Framework [2] is used to interface with the UML diagrams, and the Standard Widget Toolkit [7] is used to provide additional graphical user interface where needed. It should be noted that the SMT is continuously under development.

The rest of this paper is structured as follows: Section 2 provides an overview on the methodology, Section 3 presents and illustrates the use of the Security Modeling Tool, and Section 4 summarizes and proposes ideas for future work.

2 Methodology Overview

The methodology can be roughly divided into four phases: preparation, analysis, design, and implementation. Fig. 1 presents the methodology scope in terms of the tasks (presented in ellipse) to be performed in each phase and the generated artifacts (presented in rectangle). The preparation phase occurs at the organizational level, whereas the other three occur at the application development level.

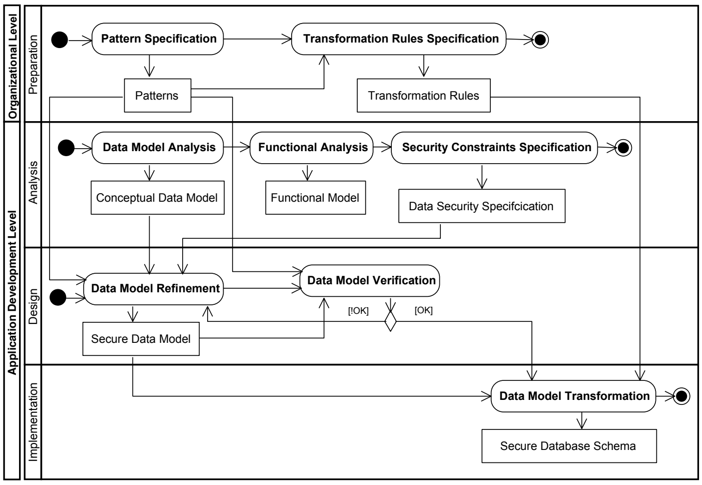


Fig. 1. Methodology overview

At the organizational level, also termed the preparation phase, we define security policies in the form of security patterns. These security patterns provide general access control policies within the organization. Once the patterns are specified, the transformation rules are defined, depicting how to transform a logical model, based on the pattern, into a database schema. The artifacts created in this phase are reusable and may be applied to various applications.

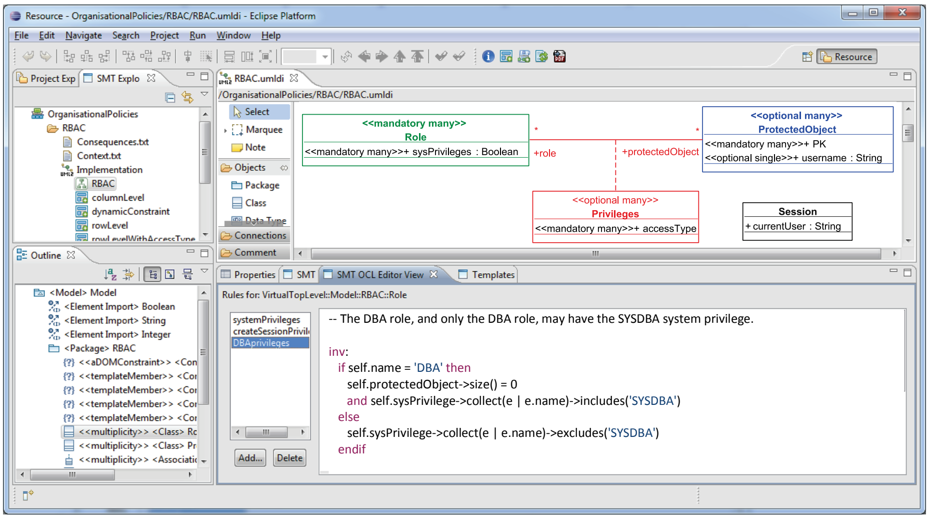
The application level deals with the development of any application within the organization. In the analysis phase of the development of an application, two models are defined, according to the FOOM methodology [6]: a conceptual data model in the form of an initial class diagram, and a functional model in the form of extended use cases. Then, the security constraints, regarding authorization to access the database, are specified. In the design phase, the artifacts from the preparation and analysis stage are used to refine the data model and create a secure data model. Next, the secure data model is verified. If the verification fails, the data model is refined until it adheres to the rules of the security patterns. In the implementation phase, the secure data model is transformed into a secure database schema. This process is defined in the transformation rules, guided by the knowledge captured in the security patterns.

3 The Security Modeling Tool

3.1 Organizational Level - The Preparation Phase

During the preparation phase, security patterns along with their transformation rules are specified. These patterns will serve as guidelines for application developers as well as a verification template, and they provide the infrastructure for the transformation process. The transformation rules depict how an application model will be transformed into a secure database schema.

Security Pattern Specification. Similarly to the classical pattern approach, security patterns are specified in a structured form. The standard template aids designers, who are not security experts, to identify and understand security problems and solve them efficiently. In order to specify the patterns, we use a common template introduced by Schumacher [5]. The template consists of five main sections: *name, context, problem, solution,* and *consequence*. The *name*, *context*, *problem*, and *consequence* sections are documentation text files. They provide the *name* of the pattern, the *context* in which the security problem occurs, the description of the security *problem*, and the *consequences* of this solution. The *solution* section provides a generic solution to the problem. It is specified with a UML class diagram that provides the static structure of the solution. In addition, OCL constraints are used to provide additional information that is inexpressible in the diagrams. Fig. 2 presents the structure of a simple Role-Based Access Control (RBAC) pattern (upper side) and an example of an OCL rule (lower side). In this example, the OCL rule restricts the number of roles that can have the SYSDBA system privilege to one, and that is the DBA role. In case that a finer grained solution is required, OCL rules in the form of general templates [8] can be defined. These general templates are specified using the specific elements that were already defined by the class diagrams specifying the structure of the pattern. In the RBAC example, the *Role*, *ProtectedObject*, *accessType* are some of those elements. The templates are essentially exemplars of the desired output code with "blanks" that should be filled in with a value of an attribute. These "blanks" contain meta-code and are delimited between "< >". After the missing values are inserted, a template engine is used to create the output code. Fig. 3 presents the instance level template that is used to specify access constraints on an instance of an object (or a row of a table in terms of relational database). The templates are used to specify fine grained access control policies during the application modeling. The developers need only to fill in the missing parameters that are inside the triangle brackets and do not need to write code in PL/SQL unless they want to express some complex constraint.

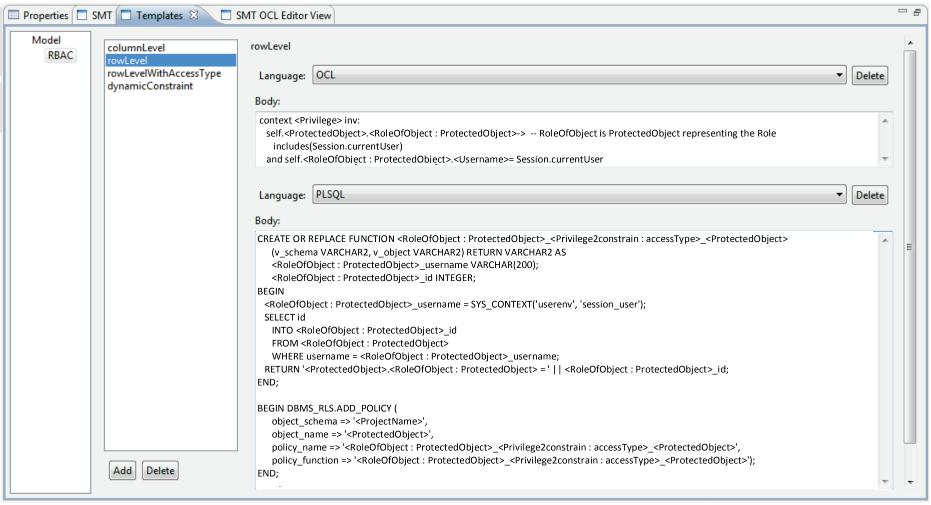


Rules for: VirtualTopLevel::Model::RBAC::Role

```
-- The DBA role, and only the DBA role, may have the SYSDBA system privilege.

inv:
  if self.name = 'DBA' then
    self.protectedObject->size() = 0
    and self.sysPrivilege->collect(e | e.name)->includes('SYSDBA')
  else
    self.sysPrivilege->collect(e | e.name)->excludes('SYSDBA')
  endif
```

Fig. 2. Pattern specification window



rowLevel

Language: OCL

Body:

```
context <Privilege> inv:
  self.<ProtectedObject>.<RoleOfObject : ProtectedObject>->  -- RoleOfObject is ProtectedObject representing the Role
    includes(Session.currentUser)
  and self.<RoleOfObject : ProtectedObject>.<Username>= Session.currentUser
```

Language: PLSQL

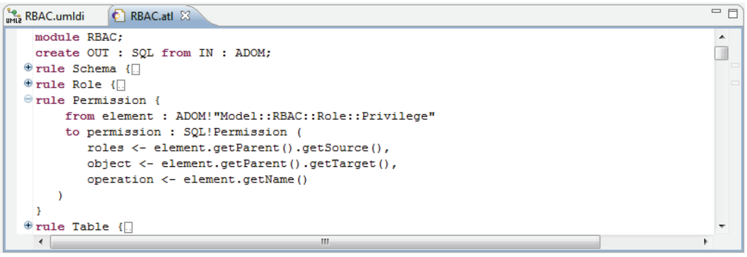
Body:

```
CREATE OR REPLACE FUNCTION <RoleOfObject : ProtectedObject>_<Privilege2constrain : accessType>_<ProtectedObject>
    (v_schema VARCHAR2, v_object VARCHAR2) RETURN VARCHAR2 AS
    <RoleOfObject : ProtectedObject>_username VARCHAR(200);
    <RoleOfObject : ProtectedObject>_id INTEGER;
BEGIN
  <RoleOfObject : ProtectedObject>_username = SYS_CONTEXT('userenv', 'session_user');
  SELECT id
    INTO <RoleOfObject : ProtectedObject>_id
    FROM <RoleOfObject : ProtectedObject>
    WHERE username = <RoleOfObject : ProtectedObject>_username;
  RETURN '<ProtectedObject>.<RoleOfObject : ProtectedObject> = ' || <RoleOfObject : ProtectedObject>_id;
END;

BEGIN DBMS_RLS.ADD_POLICY (
    object_schema => '<ProjectName>',
    object_name => '<ProtectedObject>',
    policy_name => '<RoleOfObject : ProtectedObject>_<Privilege2constrain : accessType>_<ProtectedObject>',
    policy_function => '<RoleOfObject : ProtectedObject>_<Privilege2constrain : accessType>_<ProtectedObject>');
END;
```

Fig. 3. The Instance (Row) Level Template

Transformation Rules Specification. To transform the UML class diagram into SQL code, we use the ATLAS Transformation Language (ATL) [3]. First, we transform the application model into a SQL application model, and then we translate it to SQL code. Fig. 4 shows the *transformation* rule for *Privilege*.



```
module RBAC;
create OUT : SQL from IN : ADOM;
rule Schema {
rule Role {
rule Permission {
    from element : ADOM!"Model::RBAC::Role::Privilege"
    to permission : SQL!Permission (
       roles <- element.getParent().getSource(),
       object <- element.getParent().getTarget(),
       operation <- element.getName()
   )
}
rule Table {
```

Fig. 4. ATL transformation code for the Privilege association class

3.2 Application Development Level

The Analysis Phase. The first task in the analysis phase is to create a conceptual data model based from the users' requirements. The conceptual data model is an initial class diagram that consists of data classes, their attributes and various types of relationships. Fig. 5 depicts the initial (UML) class diagram of a university registration system.

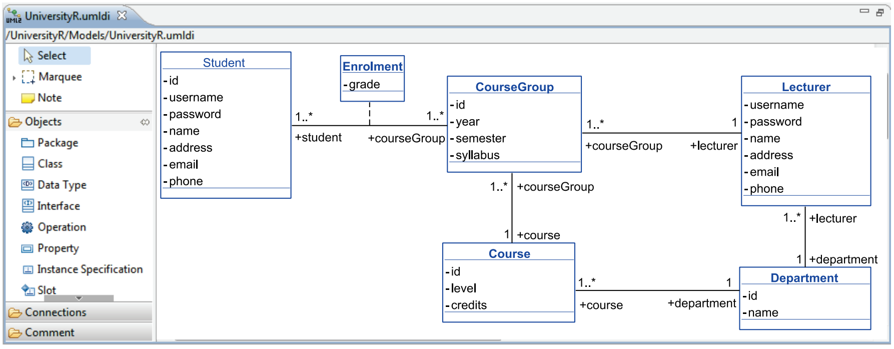


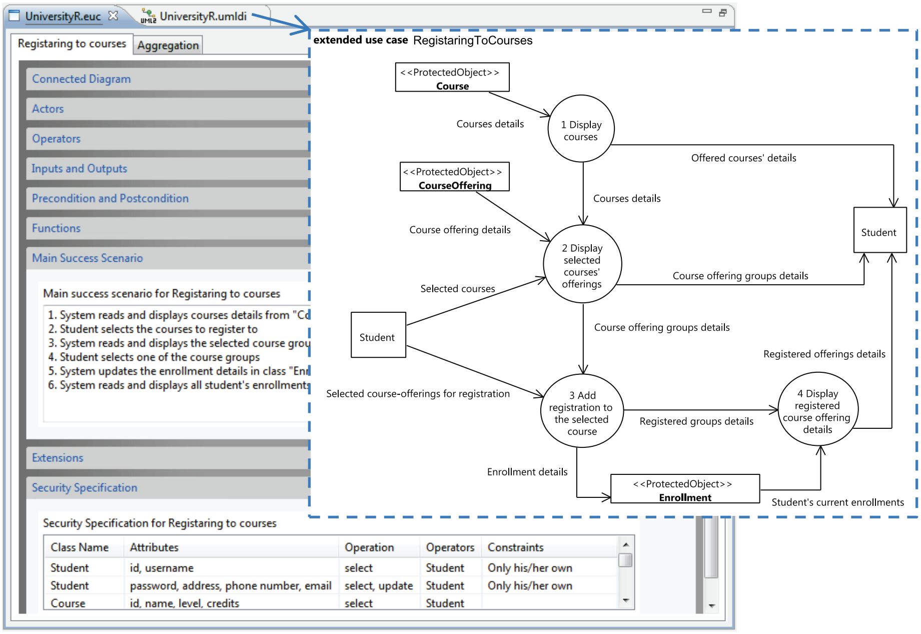
Fig. 5. An example of an initial class diagram

Following that, the functional model of the application is defined using extended use cases (EUC). An EUC is similar to a FOOM transaction [6] as it includes, besides the use case (UC) bubbles (i.e., functions), external/user entities and data classes. An external/user entity provides input data or obtains output information from the system. (It is different from an Actor in ordinary UC, which only signify who operates the use case.) Data classes, which are taken from the initial class diagram, are manipulated (i.e., retrieved or updated) by the functions of the EUC.

The EUCs enable the analyst to define, among other things, the roles of the various users of the EUC and their access privileges. As in ordinary UC, for every EUC diagram we also prepare a description. The template for an EUC description is extended compared to an ordinary UC description, as it includes definitions of access authorization.

For each class included in an EUC we define: a) the roles, i.e. the authorized operators of the EUC; b) the type of access authorization (e.g., add, read, update or delete); and c) the attributes involved in that operation. Fig. 6 shows an example of an EUC with definition of access authorization. The right side shows a certain EUC diagram; the left side shows part of the EUC description; the bottom shows a part of the security specifications: in a form of a table we show, for each class that participates in the EUC, the security specifications.

Eventually, all the security specifications, defined for all the EUCs are aggregated in one table.


UniversityR.euc
UniversityR.umldi
Registaring to courses
Aggregation
Connected Diagram
Actors
Operators
Inputs and Outputs
Precondition and Postcondition
Functions
Main Success Scenario
Extensions
Security Specification
extended use case RegistaringToCourses
<<ProtectedObject>> Course
Courses details
1 Display courses
Offered courses' details
<<ProtectedObject>> CourseOffering
Courses details
Course offering details
Student
2 Display selected courses' offerings
Selected courses
Course offering groups details
Student
Course offering groups details
Registered offerings details
Selected course-offerings for registration
3 Add registration to the selected course
Registered groups details
4 Display registered course offering details
Enrollment details
<<ProtectedObject>> Enrollment
Student's current enrollments

Main success scenario for Registaring to courses

1. System reads and displays courses details from "Co
2. Student selects the courses to register to
3. System reads and displays the selected course grou
4. Student selects one of the course groups
5. System updates the enrollment details in class "Enr
6. System reads and displays all student's enrollments

Security Specification for Registaring to courses

Class Name	Attributes	Operation	Operators	Constraints
Student	id, username	select	Student	Only his/her own
Student	password, address, phone number, email	select, update	Student	Only his/her own
Course	id, name, level, credits	select	Student	

Fig. 6. Example of an EUC with definition of access authorization

The Design Phase. During this phase, the initial class diagram is refined, by the designer, to include the security specification, adhering with the organizational policies. Namely, the authorization rules are added to the initial class diagram as specified by the patterns, and the relevant elements are classified via stereotypes according to the predefined patterns, as presented in Fig. 7. During this task, additional changes to authorization rules may be applied and fine grained restrictions may be specified using the templates that were defined in the patterns. Fig. 8 illustrates the use on the instance (row) level template that was defined in Fig. 3. To use the template, the designer merely instantiates it and provides the missing parameters. In the SMT, these parameters are listed at the bottom of the view. The SMT also provides a preview of the templates after the missing parameters were specified, for instance in OCL and PL/SQL.

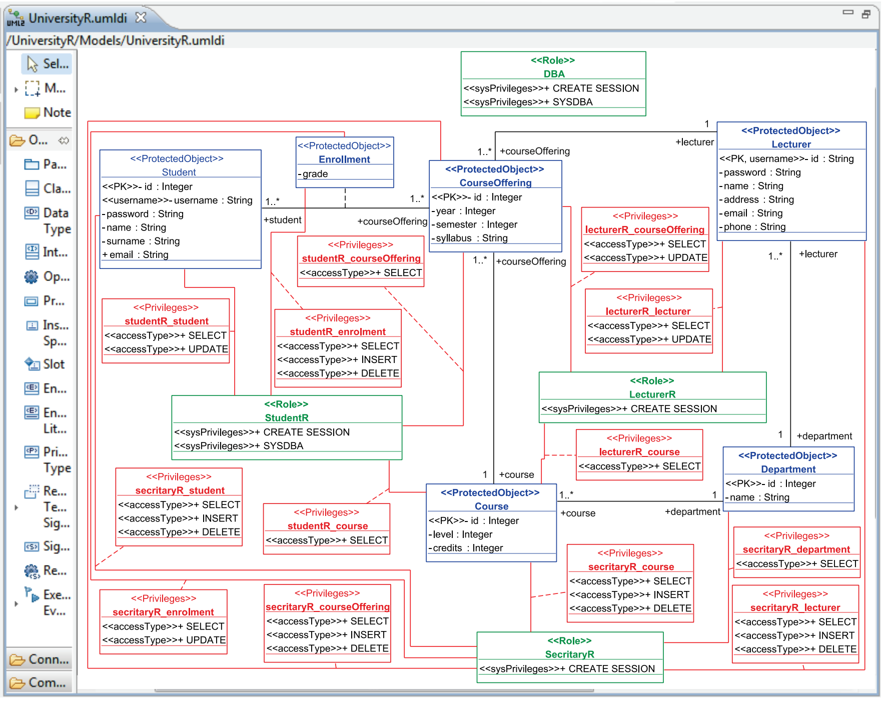
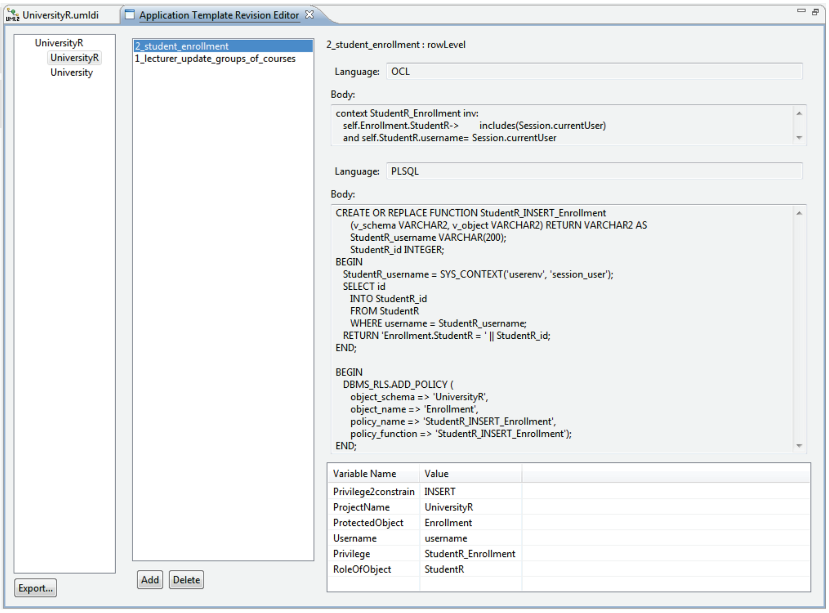


Fig. 7. An example of the RBAC-base refined data model



2_student_enrollment : rowLevel

Language: OCL

Body:

```
context StudentR_Enrollment inv:
  self.Enrollment.StudentR->      includes(Session.currentUser)
  and self.StudentR.username= Session.currentUser
```

Language: PLSQL

Body:

```
CREATE OR REPLACE FUNCTION StudentR_INSERT_Enrollment
    (v_schema VARCHAR2, v_object VARCHAR2) RETURN VARCHAR2 AS
    StudentR_username VARCHAR(200);
    StudentR_id INTEGER;
BEGIN
  StudentR_username = SYS_CONTEXT('userenv', 'session_user');
  SELECT id
    INTO StudentR_id
    FROM StudentR
    WHERE username = StudentR_username;
  RETURN 'Enrollment.StudentR = ' || StudentR_id;
END;

BEGIN
  DBMS_RLS.ADD_POLICY (
    object_schema => 'UniversityR',
    object_name => 'Enrollment',
    policy_name => 'StudentR_INSERT_Enrollment',
    policy_function => 'StudentR_INSERT_Enrollment');
END;
```

Variable Name	Value
Privilege2constrain	INSERT
ProjectName	UniversityR
ProtectedObject	Enrollment
Username	username
Privilege	StudentR_Enrollment
RoleOfObject	StudentR

Fig. 8. An example of instance level (row) constraint

After creating a refined data model, we need to check if it adheres to the security policies as defined by the specified security patterns. The SMT provides automatic verification that is based on the ADOM validation algorithm. If the application is invalid, an error message is presented, explaining the verification errors. An example of two such errors are: 1) multiplicity error: access type is not specified to the *Privilege* class *StudentR_CourseOffering*, and 2) OCL error: *StudentR* role has *the SYSDBA* privilege.

The Implementation Phase. During this phase the transformation rules, which were defined during the preparation phase, are used to translate the verified application model into a database schema.

4 Summary

We have presented the Security Modeling Tool, which supports the development of secured database schemata upon the methodology we have developed. This tool utilizes security patterns for enforcing security on database application design. The tool guides developers on how to incorporate security aspects, in particular authorization, within the development process with pre-defined security patterns. It handles the specification and implementation of the authorization aspect from the early stages of the development process, leading to a more secure system design.

Currently, we are in a process of applying the methodology along with its supporting tool within an industrial environment. This will enable us to introduce improvements in the methodology and the tool. In future work, we plan to enrich the methodology and tool to support other non-functional requirements (e.g., in the security era it might include privacy, encryption, and auditing). In addition, we plan to apply the methodology to the code level, similarly to the way we apply it to database schemata; yet, we intend to incorporate behavioral specification as well.

References

1. Dhillon GS. Information Security Management: Global Challenges in the New Millennium. IGI Publishing (2001)
2. Eclipse Modeling Framework (2011). http://www.eclipse.org/modeling/emf/
3. Jouault F, Allilaire F, Bézivin J, Kurtev I. ATL: A model transformation tool. Science of Computer Programming. 72(1-2), 31--39 (2008)
4. Reinhartz-Berger, I., Sturm, A.: Utilizing Domain Models for Application Design and Validation. Information & Software Technology 51 (8), 1275--1289 (2009)
5. Schumacher, M.: Security Engineering with Patterns: Origins, Theoretical Models, and New Applications. Springer-Verlag New York, Inc., Secaucus (2003)
6. Shoval, P.: Functional and Object-Oriented Analysis and Design - An Integrated Methodology. IGI Publishing, Hershey (2007)
7. Standard Widget Toolkit (2011). http://www.eclipse.org/swt/
8. StringTemplate (2011). http://www.stringtemplate.org/

A Tool for Managing Evolving Security Requirements*

Gábor Bergmann[1], Fabio Massacci[2], Federica Paci[2],
Thein Tun[3], Dániel Varró[1], and Yijun Yu[3]

[1] DMIS - Budapest University of Technology and Economics,
{bergmann,varro}@mit.bme.hu
[2] DISI - University of Trento,
{fabio.massacci,federica.paci}@unitn.it
[3] DC - The Open University
{t.t.tun,y.yu}@open.ac.uk

Abstract. Requirements evolution management is a daunting process. Requirements change continuously making the traceability of requirements hard and the monitoring of requirements unreliable. Moreover, changing requirements might have an impact on the security properties a system design should satisfy: certain security properties that are satisfied before evolution might no longer be valid or new security properties need to be satisfied. This paper presents SeCMER, a tool for requirements evolution management developed in the context of the SecureChange project. The tool supports automatic detection of requirement changes and violation of security properties using change-driven transformations. The tool also supports argumentation analysis to check security properties are preserved by evolution and to identify new security properties that should be taken into account.

Keywords: security requirements engineering, secure i*, security argumentation, change impact analysis, security patterns

1 Introduction

Modern software systems are increasingly complex and the environment where they operate is increasingly dynamic. The number and needs of stakeholders is also changing constantly as they need to adjust to the changing environment. A consequence of this trend is that the requirements for a software system increases and changes continually. To deal with evolution, we need analysis techniques that assess the impact of system evolution on the satisfaction of requirements such as security of the system which is very sensitive to evolution: security properties satisfied before evolution might no longer hold or new security properties need to be satisfied as result of the evolution.

Another important aspect is the change management process itself which is a major problem in practice. Changes make the traceability of requirements

* Work parly supported by the project EU-FP7-ICT-FET-IP-SecureChange.

hard and the monitoring of requirements unreliable: requirements management is difficult, time-consuming and error-prone when done manually. Thus, a semi-automated requirements evolution management environment, supported by a tool, will improve requirement management with respect to keeping requirements traceability consistent, realizing reliable requirements monitoring, improving the quality of the documentation, and reducing the manual effort.

In this paper we present SeCMER[4], a tool developed in the context of the SecureChange European project[5]. The tool supports the different steps of SeCMER methodology for evolutionary requirements [4]. The methodology allows to model requirement evolution in different state of the art requirement languages such as SI* [6], Problem Frames (PF) [9] and SeCMER that is a requirement language that includes concepts belonging to SI*, PF and security such as asset. The methodology also supports the automatic detection of requirement changes and violation of security properties and argumentation analysis [9] to check security properties are preserved by evolution and to identify new security properties that should be taken into account. Change driven transformations based on *evolution rules* [3] are leveraged to check argument validity, to automatically detect violations or fulfilment of security properties, and to issue alerts prompting human intervention, a manual analysis or argumentation process, or trigger automated reactions in certain cases.

In the next section (§2) we describe the tool architecture, then we illustrate the tool features based on an industrial example of evolution taken from the air traffic management domain (§3) and finally conclude the paper (§4).

2 SeCMER Tool architecture

SeCMER is an Eclipse-based heterogeneous modeling environment for managing evolving requirements models. It has the following features (See also Fig. 1):

- **Modeling of Evolving Requirements**. Requirement models can be drawn in SI*, Problem Frames or SeCMER. Traceability and bidirectional synchronization is supported between SeCMER and SI* requirements models.
- **Change detection based on evolution rules**. Violations of formally defined static security properties expressed as patterns can be automatically identified. Detection of formal or informal arguments that has been invalidated by changes affecting model elements that contributed to the argument as evidence is also supported.
- **Argumentation-based security analysis**. Reasoning about security properties satisfaction and identification of new security properties is supported.

These capabilities of the tool are provided by means of the integration of a set of EMF-based [7] Eclipse plug-ins written in Java, relying on standard EMF technologies such as GMF, Xtext and EMF Transaction.

[4] A detailed description of the tool implementation is reported in [5]

[5] www.securechange.eu

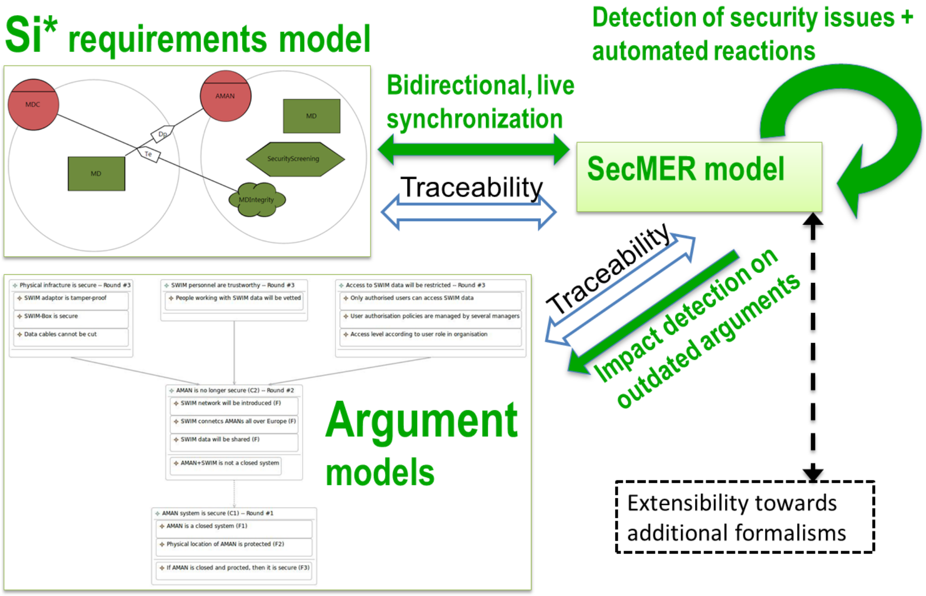

Si* requirements model
MDC
AMAN
MD
Dp
Te
MD
SecurityScreening
MDIntegrity
Bidirectional, live synchronization
Traceability
Detection of security issues + automated reactions
SecMER model
Traceability
Impact detection on outdated arguments
Extensibility towards additional formalisms
Argument models
Physical infracture is secure -- Round #3
SWIM adaptor is tamper-proof
SWIM-Box is secure
Data cables cannot be cut
SWIM personnel are trustworthy -- Round #3
People working with SWIM data will be vetted
Access to SWIM data will be restricted -- Round #3
Only authorised users can access SWIM data
User authorisation policies are managed by several managers
Access level according to user role in organisation
AMAN is no longer secure (C2) -- Round #2
SWIM network will be introduced (F)
SWIM connetcs AMANs all over Europe (F)
SWIM data will be shared (F)
AMAN+SWIM is not a closed system
AMAN system is secure (C1) -- Round #1
AMAN is a closed system (F1)
Physical location of AMAN is protected (F2)
If AMAN is closed and procted, then it is secure (F3)

Fig. 1. Models and features in the SeCMER tool

SeCMER integrates Si* [6] as a graphical modeling framework for security requirements, with OpenPF [8] that supports formal and informal manual argumentation of security properties. Change detection for security patterns and evolution rules, as well as the detection of invalidated arguments are performed using EMF-IncQuery [2].

The core trigger engine plug-in offers an Eclipse extension point for defining change-driven rules. Multiple constituent plug-ins contribute extensions to register their respective set of rules. The graph pattern-based declarative event/condition feature of the rules is evaluated efficiently (see measurements in [2]) by the incremental graph pattern matcher plug-ins automatically generated from the declarative description by EMF-IncQuery. At the commit phase of each EMF transaction, the rules that are found to be triggered will be executed to provide their reactions to the preceding changes. These reactions are implemented by arbitrary Java code, and they are allowed to modify the model as well (wrapped in nested transactions) and could therefore be reacted upon.

So far, there are three groups of change-driven rules as extension points:

- transformation rules that realize the on-the-fly synchronization between multiple modeling formalisms,
- security-specific evolution rules that detect the appearance of undesired security patterns, raise alerts and optionally offer candidate solutions.
- rules for invalidating arguments when their ground facts change.

A major feature is the a bi-directional synchronizing transformation between Si* and the SeCMER model with changes propagated on the fly, interactively. Since the languages have different expressive power, the following challenges arise:

1. some concepts are not mapped from one formalism to the other or vice versa,
2. some model elements may be mapped into multiple (even an unbounded amount of) corresponding model elements in the other formalism, and finally
3. it is possible that a single model element has multiple possible translations (due to the source formalism being more abstract); one of them is created as a default choice, but it can later be changed to the other options, which are also tolerated by the transformation system.

3 Demo Scenario

We are going to illustrate the features supported by our prototype using the ongoing evolution of ATM systems as planned by the ATM 2000+ Strategic Agenda [1] and the SESAR Initiative.

Part of ATM system's evolution process is the introduction of the Arrival Manager (AMAN), which is an aircraft arrival sequencing tool to help manage and better organize the air traffic flow in the approach phase. The introduction of the AMAN requires new operational procedures and functions that are supported by a new information management system for the whole ATM, an IP based data transport network called System Wide Information Management (SWIM) that will replace the current point to point communication systems with a ground/ground data sharing network which connects all the principal actors involved in the Airports Management and the Area Control Centers.

The entities involved in the simple scenario used for this demo are the AMAN, the Meteo Data Center (MDC), the SWIM-Box and the SWIM-Network. The SWIM-Box is the core of the SWIM information management system which provides access via defined services to data that belong to different domain such as flight, surveillance, meteo, etc. The introduction of the SWIM requires suitable security properties to be satisfied: we will show how to protect information access on meteo data and how to ensure integrity of meteo data.

1. **Requirements evolution**. We show how SeCMER supports the representation of the evolution of the requirement model as effect of the introduction of the SWIM.
2. **Change detection based on evolution rules.**
   a *Detection of a security property violation based on security patterns.* We show how the tool detects that the integrity security property of the resource MD "Meteo Data" is violated due to the lack of a trusted path.
   b *Automatically providing corrective actions based on evolution rules.* We show how evolution rules may suggest corrective actions for the detected violation of the integrity security property.
3. **Argumentation-based security analysis**. We show how argumentation analysis [9] can be carried to provide evidence that the information access property applied to the meteo data is satisfied after evolution.

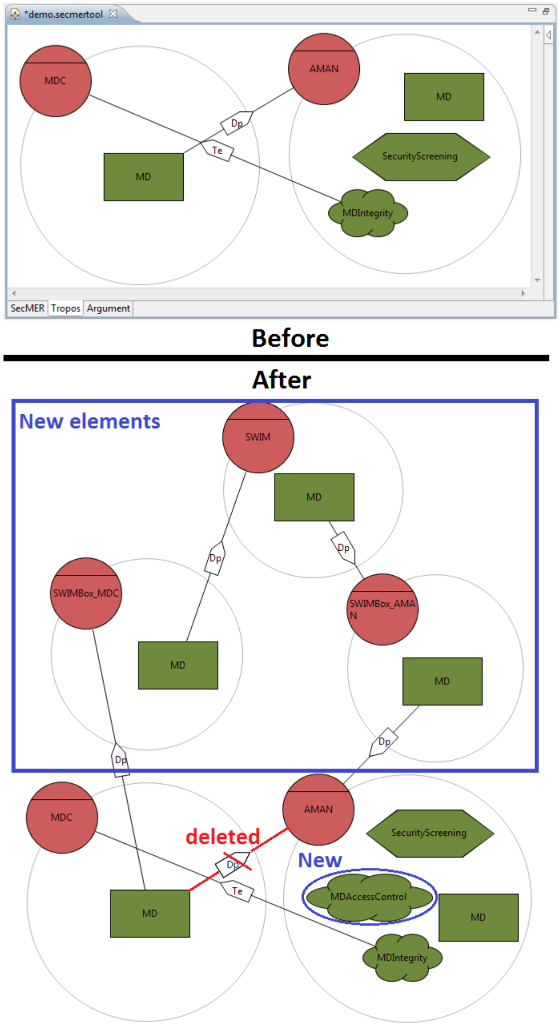


Fig. 2. Annotated screenshot fragments showing requirements evolution

The steps of the demo were chosen such that they follow a typical requirements evolution workflow, also featuring the contributions of WP3.

Requirements evolution. Fig. 2 shows the evolution of the model. The before model includes two actors the *AMAN* and *MDC*: MDC provides the asset Meteo Data (*MD*) to the AMAN. The AMAN has an integrity security goal *MDIntegrity* for MD, and MDC is entrusted with this goal. AMAN also performs an Action, *SecurityScreening*, to regularly conduct a background check on its employees to ensure that they do not expose to risk the information generated by the AMAN.

Listing 1 Pattern to capture violations of the trusted path property

```
shareable pattern
noTrustedPath(ConcernedActor,SecGoal,Asset,UntrustedActor)={
  Actor.wants(ConcernedActor,SecGoal);
  SecurityGoal(SecGoal);
  SecurityGoal.protects(SecGoal, Asset);
  Actor.provides(ProviderActor,Asset);
  find
transitiveDelegation(ProviderActor,UntrustedActor,Asset);
  neg Actor.trust*(ConcernedActor,UntrustedActor);
  neg find
trustedFulfillment(ConcernedActor,AnyActor,AnyTask,SecGoal); }
```

As the communication between the AMAN and MDC is mediated by the SWIM, the before model evolves as follows:

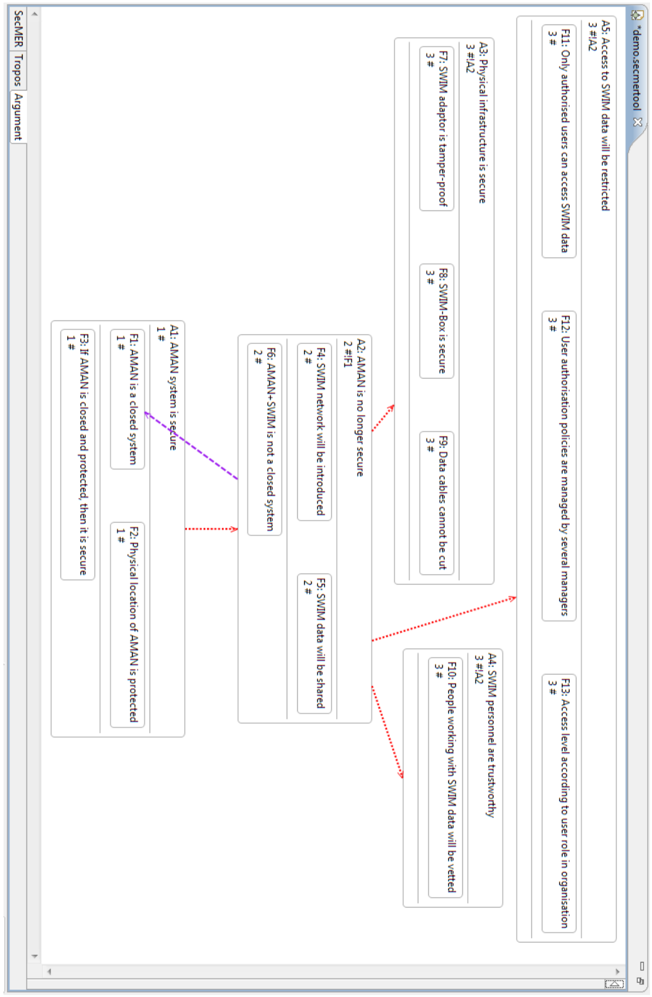
- The Actors *SWIM*, *SWIMBox_MDC* and *SWIMBox_AMAN* are introduced in the SI* model
- As the meteo data is no longer directly provided by MDC to AMAN, the delegation relation between the two is removed.
- Delegation relationships are established between the Actors MDC, SWIMBox_MDC, SWIM, SWIMBox_AMAN, AMAN.
- As the SWIM network can be accessed by multiple parties, the AMAN has a new security goal *MDAccessControl* protecting MD resource.

Detecting violations of security properties based on security patterns. SeCMER includes facilities that allow for the declarative definition of security patterns that express situations that leads to the violation of a security property. For example, if a concerned actor wants a security goal that expresses that a resource must be protected, then each actor that the resource is delegated to must be trusted (possibly transitively) by the concerned actor. An exception is made if a trusted actor performs an action to explicitly fulfill the security goal, e.g. digital signature makes the trusted path unneccessary in case of an integrity goal. See Lst. 1 for the definition of the pattern using the declarative model query language of EMF-IncQuery [2].

According to this pattern the integrity property for MD is violated because AMAN entrusts MDC with the integrity security goal, but the communitation intermediary actors SWIMBox_MDC, and SWIMBox_AMAN are not.

Automatic corrective actions based on evolution rules. The security pattern in Lst. 1 can be used to define evolution rules that define automated corrective actions to be applied to the model in order to re-establish the integrity security property. Possibile examples of corrective actions are:

- Add a trust relationship between MDC and SWIM Network having the integrity security goal as dependum.
- Alternatively, an Action such as MD is digitally signed can be created to protect the integrity of MD even when handled by untrusted actors.



A5: Access to SWIM data will be restricted
3 #A2

F11: Only authorised users can access SWIM data
3 #

F12: User authorisation policies are managed by several managers
3 #

F13: Access level according to user role in organisation
3 #

A3: Physical infrastructure is secure
3 #A2

F7: SWIM adaptor is tamper-proof
3 #

F8: SWIM-Box is secure
3 #

F9: Data cables cannot be cut
3 #

A4: SWIM personnel are trustworthy
3 #A2

F10: People working with SWIM data will be vetted
3 #

A2: AMAN is no longer secure
2 #F1

F4: SWIM network will be introduced
2 #

F5: SWIM data will be shared
2 #

F6: AMAN+SWIM is not a closed system
2 #

A1: AMAN system is secure
1 #

F1: AMAN is a closed system
1 #

F2: Physical location of AMAN is protected
1 #

F3: If AMAN is closed and protected, then it is secure
1 #

Fig. 3. Screenshot fragment showing the argumentation model

Argumentation for the information access property. Fig. 3 shows the different rounds of the argumentation analysis that is carried out for the infomation access security property applied to MD resource.

The diagram says that the AMAN system is claimed to be secure before the change (Round #1), and the claim is warranted by be the facts the system is known to be a close system (F1), and the physical location of the system is protected (F2). This argument is rebutted in Round #2, in which another argument claims that the system is no longer secure because SWIM will not keep AMAN closed. The rebuttal argument is mitigated in Round #3 by three arguments, which suggest that the AMAN may still be secure given that the physical infrastructure is secure, personnel are trustworthy and access to data is controlled.

4 Conclusions

The paper presented SeCMER, a tool for managing evolving requirements. As shown by the ATM-based demo scenario, the tool supports visual modeling of security requirements. Additionally, argument models can be constructed manually to investigate the satisfaction of security properties; the tool detects invalidated arguments if the requirements model evolves. Finally, the tool performs continuous and automatic pattern-based security properties violation detection, with "quick fix" corrective actions specified by evolution rules.

We are planning to extend the tool in order to support other set of security patterns and evolution rules to automate the detection and handling of security violations in a wider range of application scenarios. We will also realize a tighter integration with additional modeling formalisms (Problem Frames) and industrial tools e.g DOORS-TREK. The usability and the features of the tool are going to be evaluated through a study involving ATM-domain experts.

References

1. EUROCONTROL ATM Strategy for the Years 2000+ Executive Summary (2003)
2. Bergmann, G., et al.: Incremental evaluation of model queries over EMF models. In: Model Driven Engineering Languages and Systems, MODELS'10. Springer (2010)
3. Bergmann, G., et al.: Change-Driven Model Transformations. Change (in) the Rule to Rule the Change. Software and System Modeling (2011), to appear.
4. Bergmann et al.: D3.2 Methodology for Evolutionary Requirements, http://www.securechange.eu/sites/default/files/deliverables/D3.2-%20Methodology%20for%20Evolutionary%20Requirements_v3.pdf
5. Bergmann et al.: D3.4 Proof of Concept Case Tool, http://www.securechange.eu/sites/default/files/deliverables/D3.4%20Proof-of-Concept%20CASE%20Tool%20for%20early%20requirements.pdf
6. Massacci, F., Mylopoulos, J., Zannone, N.: Computer-aided support for secure tropos. Automated Software Engg. 14, 341–364 (September 2007)
7. The Eclipse Project: Eclipse Modeling Framework, http://www.eclipse.org/emf
8. Tun, T., et al.: Early identification of problem interactions: A tool-supported approach. In: Glinz, M., Heymans, P. (eds.) Requirements Engineering: Foundation for Software Quality, 15th International Working Conference, pp. 74–88. No. 5512 in Lecture Notes in Computer Science, Springer (2009)
9. Tun, T.T., et al.: Model-based argument analysis for evolving security requirements. In: Proceedings of the 2010 Fourth International Conference on Secure Software Integration and Reliability Improvement. pp. 88–97. SSIRI '10, IEEE Computer Society, Washington, DC, USA (2010)

A Software Framework for the Automated Production of Schematic Maps

Joao Mourinho[1], Teresa Galvao[1], Joao Falcao e Cunha[1], Fernando Vieira[2], and Jose Pacheco[2]

[1] Faculdade de Engenharia da Universidade do Porto, 4200-465 Porto
[2] OPT - Optimizacao e Planeamento de Transportes, 4200 Porto
joao.mourinho@fe.up.pt

Abstract. Schematic Maps are mainly used for depicting transportation networks. They are generated through a schematization process where irrelevant details are eliminated and important details are emphasized. This process, being manually performed by teams of expert designers, is expensive and time consuming. Such manual execution is unsuitable for the production of schematic maps for location-based services or on-demand schematic maps, as near real-time and user-centered properties are needed. This work proposes GeneX, a framework that can support the automated generation of schematic maps. The framework and the new algorithms developed were able to completely eliminate erroneous map point placement, and to decrease by 33% the contention for map point placement, producing schematic maps without human intervention in soft real time.

Keywords: Schematic Maps, Software Framework, Public Transportation

1 Introduction

Schematic maps have been increasingly used in response to the need of better and simpler maps to describe complex transport networks. This apparent simplicity is achieved through a simplification process called "schematization process" where choices are made regarding the level of detail and simplification. A special type of schematic map, called spider map, has also appeared recently. It presents innovative features such as a spider structure improve visual presentation, user learning and spatial context communication. Schematic maps, by their inherent simplicity and symbolic meaning are good maps for being used in the transportation area as they are far more intuitive than conventional maps [1]. In fact as people travel more often, they need flexible and easy to understand maps which may take in account their context [2]. Automation in the production of maps is a key factor to achieve flexibility to tailor maps to user context, as it happens with Location-Based Services [3]. There is the need, then, to develop a software framework which could support efficiently and comprehensively the automated generation of schematic maps. In this paper we propose and describe a software framework which serves as an engine to the generation and test of schematic maps.

2 Schematic and Spider Maps

Some authors define schematic map as *"an easy-to-follow diagrammatic representation based on highly generalized lines which is in general used for showing routes of transportation systems, such as subways, trams and buses, or for any scenario in which streams of objects at nodes in a network play a role"* [4]. One remarkable schematic map applied to a transportation network was the Harry Beck's London Underground diagram. Beck's map was considered both bold and innovative, as for the first time lines were drawn either horizontally, vertically or diagonally at 45. This map also uses differential zoom scaling and although it gives the traveler some clues about the terrain features (ex: river) and his/her location, it does not mimic the geography of London. Spider maps are special schematic maps. Like schematic maps, the stops and lines of the transportation network correspond to vertices and edges. However, they have enhanced features such a spider architecture, thus having a specific set of characteristics which sets them apart from schematic maps. Spider maps pay special attention to context in order to enhance user learning and ease of use. A spider map such as the one depicted in figure 1, comprehends three components:

- **Hub:** Describes the area in which the user is, as well as the surrounding area with a higher degree of detail (buildings, roads, etc). The hub, as it is the central part of the spider map, is the first component the user will look at, as it makes uses of *"focus and context"* [5] and detail focusing techniques. The hub may not comply with the 0/45/90 degrees line orientation.
- **Lines:** The lines follow the orthogonal orientation of the traditional schematic maps, and describe the paths of the transport network where the user can go through while being at the zone depicted by the hub.
- **Stops:** The stops are the destinations accessible to the user from the hub.

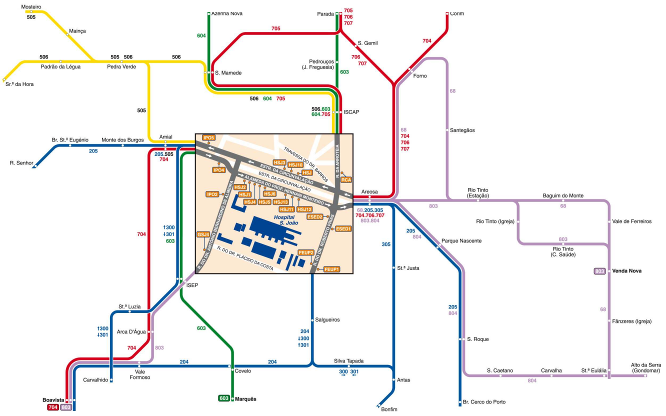


Fig. 1. Hospital de Sao Joao (Porto, Portugal) spider map. [6]

Visual simplicity of schematic maps is achieved through a sequential decision process regarding the level and nature of detail and schematization. In practice, this "schematization process" is still a manual process carried away by teams of expert designers. The automation of the schematization process requires effective and efficient algorithms to achieve, in one hand, high quality schematic maps which can be understood by people and, in the other hand, a time-efficient process. Through schematization, certain map details are emphasized while others are deemphasized. It is fundamental to present the smallest amount of information the user needs to learn the map to decrease user learning time. Therefore information shall be reduced to its basic components to achieve that goal. There are some studies regarding the automated drawing of schematic maps [7] [8] [9] [10], nevertheless these studies tend to focus only some areas of the problema and do not make a multidisciplinary approach. They mostly focus on the schematization process [11]. Nollenburg [12] [13] makes a deep research on the discrete mathematical foundations which are the basis of the algorithms used in the drawing of schematic maps and makes some brief considerations about their implementation. Nevertheless, his studies do not cover the human perception factors nor a concrete computer framework for drawing schematic maps. Silvana Avelar [1] [4] presents a broader study, by including some human perception factors and studies the schematic maps on demand. She goes further on by presenting a framework for electronic schematic maps which can answer user queries and studies the automated generation of schematic maps. Nevertheless, the study of the human perception factors is limited to what she calls the "aesthetic factors". Most of the algorithms to design schematic maps retain a common structure [14]. They use a graph to model the transportation network, in which the vertices represent stops or turning points and the edges represent the paths between two turning points.

3 The GeneX Framework

In this section, we present the GeneX framework which was developed through a collaboration research performed by a team involving collaborators from FEUP *(http://www.fe.up.pt)*, OPT *(http://www.opt.pt)*, STCP *(http://www.stcp.pt)*, FWT *(http://www.fwt.co.uk)*, INEGI *(www.inegi.up.pt)*, and that was funded by INEGI. The GeneX framework is a software application designed to support the following objectives:

1. The automatic generation of electronic schematic maps for complex transportation networks in bounded time, through the flexible use and parameterization of schematization algorithms
2. To serve as a test lab to support the research of schematic maps.

By merging and processing different kinds of external information (transportation networks, geographic and constraint information) through the use of state-of-the-art algorithms, the framework generates schematic maps automaticaly. The framework produces an SVG [3] file which can be used directly, printed in paper or further processed. The framework alsos produce a statistics file to measures several parameters about the framework functioning.

[3] The Scalable Vectorial Graphic is XML language to describe vectorial bidimentional graphics. It is an open format created by the World Wide Web Consortium

3.1 Software Engineering Life Cycle model

The requirements elicitation and validation was performed together with the project stakeholders, as part of a Joint Application Development (JAD) model [15]. The contributions provided by the partners were highly regarded: from the experts in information systems for transportation services (OPT), in optimization algorithms (INEGI) and map design (FWT), to the final users of the system (STCP). The development of this framework was considered, since the begining, an interdisciplinar subject in which knowledge from several areas of the science need to be integrated. Regarding functional requirements, the framework should be capable of producing schematic and spider maps in a fully automated way, about any location the user may select, using several schematization algorithms. The framework should be able to obtain data through the use of a common standard protocol data schema shared by the stakeholders. Another requirement was that the producton of schematic maps should be time-bounded (by setting deadlines or iterations number), to make it able to support Location-based services. In order to serve as a test lab, the framework should allow the choice of the algorithm and its parameters. The framework should also support different schematization algorithms (genetic, linear, tabu search, GRASP, etc) and provide a common protocol to implement them. This set of functional requirements needs to be supported by a set of non-funcional requirements, which comprehends usability, performance and interoperability. Usability is fundamental as the schematic maps produced by the framework have a strong user learning component: the maps produced, as well all possible interactions should be as intuitive as possible to be quickly learned and understood by their users. The designers and the final user team members provide insightful highlights in this area. To be able to support location-based services, the framework should execute the algorithm and produce the correspondent schematic map in near real time. Therefore, the framework was designed to perform as a soft real-time system [16]. A standardized transport network data specificication was implemented and a common interface for the algorithms was designed to achieve interoperability. Extensive use of reusable components [17] [18] was also made. The framework was developed by using C# Language, as a modern Object Oriented language which supported the requirement list.

3.2 Architecture and Data Model

The architecture of the framework follows a modular structure, with two main modules: the data preparation module and the algorithm execution module. Figure 2 shows the GeneX framwork package diagram, depicting its components.

The data aquisition and preparation module is responsible for preparing the data to be used by the algorithm execution module. The user selects graphically the location where he wants the spider/schematic map to be centered (hub). The module then extracts raw data from the transport network database and organizes it into a data structure by using the Spider Map Library. The spider

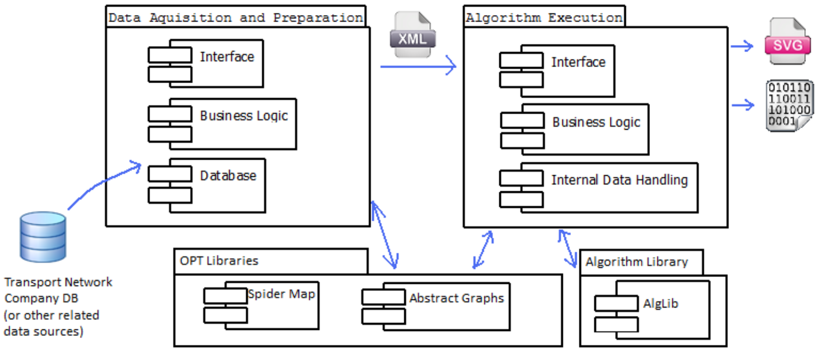


Fig. 2. GeneX Framework Package Diagram

map library is a complex set of C# classes that support the serialization and communicaton of the spider and schematic map structure. The algorithm execution module reads the XML file and transforms the data into an internal data structure (to improve component reuse by the schematization algorithms). At the user interface, the user can choose the schematization algorithm (from the AlgLib algorithms package) to execute and its options and performance measurement metrics. The business logic is then responsible for calling and executing the algorithm and to produce the final result, which can be an SVG (Scalable Vector Graphics) or a binary file containing serialized data. The SVG file is produced by using a library that allows the conversion of a spider map data structure in an SVG file called Abstract Graphs Library [6].

3.3 The HPPO Algorithm

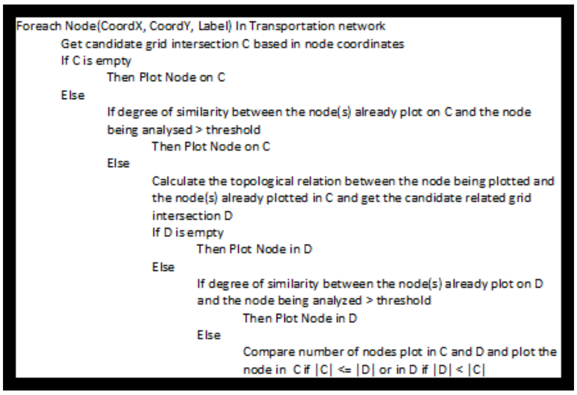
The AlgLib package provides a foundation for the execution and configuration of the schematization algorithms. Each algorithm has to implement the same communication interface functions, in order to be used by the execution module:

- ***spiderMap*** **execute(*spiderMap*, *parameterList*)** performs the execution of the algorithm. The arguments are the spider map XML structure that was opened by the execution module, and the algorithm parameter list, already set up by the user. This function returns a spiderMap data structure which is the processed spider/schematic map. That structure can then be output to a SVG or serialized binary file.
- ***parameterList*** **getParameters()** the module calls this parameter function to get the list of the algorithm parameters that can be set by the user.

We have implemented a preliminary schematization algorithm that we called "Heuristic Point Placement Optimization" (HPPO). HPPO, aligns map points (corresponding to the transportation network stops and stations) to a regular grid, by positioning each map point in the nearest grid intersection. For high density or non regular density transportation maps, map station contention for grid intersections will happen. In this case, the HPPO smartly solves the contention through an heuristic algorithm. By using regular expressions, the point labels are also taken into account when placing network vertices, by determining the similarity degree of node labels in conjunction with its geographic location in order to produce a decision about the location to plot the node where the contention arises. The use of regular expressions is based in a finite-state automaton which scans strings in order to find the degree of similarity. We are not only relying in geographical information, but merging knowledge from different science fields theory to make a higher quality judgement on how to solve the contention, such as computer science and operations research. For each map point, HPPO starts by checking if the nearest grid intersection is empty. If it is, then there is no contention and the point is plotted there. If the grid intersection is not empty, then we have a contention. This means that the geographical coordinates of the nodes are equal, or at least, within the same decision range concerning the square grid resolution. The automaton also tells us the degree of similarity. If the degree of similarity is higher than the predefined threshold level, then we assume that both nodes refer to the same location. In this case, they should be both plotted in the same grid intersection. If the degree of similarity is lower than the threshold, then it is important to distinguish them and plot them in different grid intersections while maintaining the topological relation between them. In this case, we get the topological relation between the two points (based on their coordinates) and try to move the node to the adequate grid intersection. It was found that preserving the topological relation between map points is of fundamental importance when developing map design frameworks. If contention happens again (what can happen in a highly crowded map or with a loose square grid), then we have two options: or we may continue this cycle recursively until the topological relation is violated, or we may decide if we shall plot the node into the suggested grid intersection. To limit the processing time, we discarded the recursive approach in our algorithm. Being so, we analyse the proposed grid intersection. If contention happens again we check the degree of label similarity through our automaton, and if is higher than the threshold, we plot the point there. If not, we analyse both the first and the actual grid intersections suggested and we add the node where there are less nodes plotted. The pseudoalgorithm is described in figure 3.

4 Results

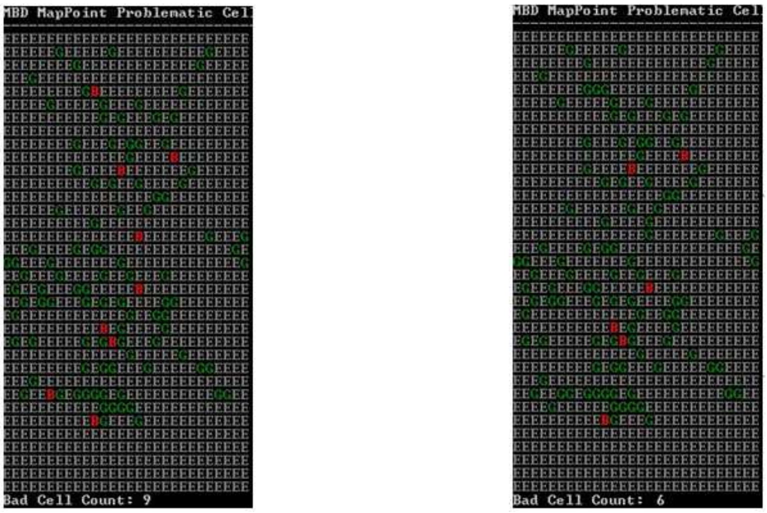
The GeneX Framework was able to generate in a fully automated way schematic and spider maps for every location requested. It is also a very important tool as a test lab for the schematization algorithms that are being developed. Although the framework is still under development and the algorithms in the AlgLib are being improved and enhanced, it is already being used for the production of schematic and spider maps. Maps produced with this framework are already available for public use in the cities of Porto and Lisbon. Concerning our HPPO algorithm, it showed good results, as it can be observed in figure 4. Other advantage of HPPO is that if different nodes refer to the same place, this algorithm can ignore



```
Foreach Node(CoordX, CoordY, Label) In Transportation network
    Get candidate grid intersection C based in node coordinates
    If C is empty
        Then Plot Node on C
    Else
        If degree of similarity between the node(s) already plot on C and the node
        being analysed > threshold
            Then Plot Node on C
        Else
            Calculate the topological relation between the node being plotted and
            the node(s) already plotted in C and get the candidate related grid
            intersection D
            If D is empty
                Then Plot Node in D
            Else
                If degree of similarity between the node(s) already plot on D
                and the node being analyzed > threshold
                    Then Plot Node in D
                Else
                    Compare number of nodes plot in C and D and plot the
                    node in C if |C| <= |D| or in D if |D| < |C|
```

Fig. 3. HPPO pseudoalgorithm description.

contention and plots them correctly in the same grid intersection (grouping). In addition, all the topological relations are still preserved.



MBD MapPoint Problematic Cel

Bad Cell Count: 9

MBD MapPoint Problematic Cel

Bad Cell Count: 6

Fig. 4. Frequency matrixes showing Bad Cells (in red) when not using HPPO(left) and using HPPO(right) for the Porto downtown spider map. HPPO reduced Bad Cells from 9 no 6, while preserving topological relations

5 Conclusions and Future Work

The GeneX framework was used to produce spider maps that are being used in the cities Porto and Lisbon, it has proved effective in generating spider maps. Nevertheless, these maps were not completely produced by an automated process, as some manual changes were necessary to improve the visual appearance, which is the most difficult aspect to model in the algorithms. The quality of the results of the HPPO algorithm is quite good, showing a significant decrease in bad cells. The algorithms need to be perfected in order to increase the quality of the results and to make them directly usable (without any manual processing). The algorithms need to further improve aspects such as visual line distiction, stop label organization and geographical constraints. Some development of the framework is also needed to support Location-based services, such a "request manager" which can feed user requests to the framework and reply to them. Other issues that need further study are the adaptation of the resulting maps to different devices. All this work also needs to be complemented and validated with usability tests and analysis.

References

1. Avelar, S.: Schematic Maps on Demand: Design, Modeling and Visualization. PhD thesis, Swiss Federal Institute of Technology Zurich (2002)
2. Porathe, T.: User-Centered Map Design. Design (2007)
3. Steiniger, S., Neun, M., Edwardes, A.: Foundations of location based services. (2006)
4. Avelar, S., Hurni, L.: On the Design of Schematic Transport Maps. Cartographica: The International Journal for Geographic Information and Geovisualization **41** (2006) 217–228
5. Bogen, S., Brandes, U., Ziezold, H.: Visual Navigation with Schematic Maps. Visual Information Communication (2010) 65–84
6. OPT: Abstract Graphs Library Specification (Internal Report) (2009)
7. Cabello, S., Deberg, M., Vankreveld, M.: Schematization of networks. Computational Geometry **30** (2005) 223–238
8. Cabello, S., Kreveld, M.V., Sciences, C., Box, P.O.: Schematic Networks: An Algorithm and its Implementation. In Richardson, D., Oosterom, P., eds.: 10th International Symposium on Spatial Data Handling (SDH), Ottawa, Springer (2002) 475–486
9. Barkowsky, T., Latecki, L.J., Richter, K.f.: Schematizing Maps : Simplification of Geographic. Cognition **8** (2000) 41–53
10. Anand, S., Avelar, S., Ware, J.M., Jackson, M.: Automated schematic map production using simulated annealing and gradient descent approaches. Technology (2000)
11. Ware, J.M., Taylor, G.E., Thomas, N.: Automated Production of Schematic Maps for Mobile Applications. Transactions in GIS **10** (2006) 25– 42
12. Nöllenburg, M.: Automated drawing of metro maps. PhD thesis, Universitat Karlsruhe (2005)
13. Nöllenburg, M., Wolff, A.: A Mixed-Integer Program for Drawing High-Quality Metro Maps. Graph Drawing **3843** (2006) 321–333
14. Dong, W., Guo, Q., Liu, J.: Schematic road network map progressive generalization based on multiple constraints. Geo-spatial Information Science **11** (2008) 215–220
15. Scacchi, W.: Process Models in Software Engineering. October (2001) 1–24
16. Laurini, R., Servigne, S., Nol, G.: Soft real-time GIS for disaster monitoring, Springer-Verlag (2005) 465–480
17. Neighbors, J.M.: The Draco approach to constructing software from reusable components. IEEE Transactions on Software Engineering **10** (1984) 567–574
18. Goguen, J.A.: Reusing and interconnecting software components. Computer **19** (1986) 16–27

DBIScholar: An iPhone Application for Performing Citation Analyses

Andreas Robecke, Ruediger Pryss, and Manfred Reichert

Institute of Databases and Information Systems, Ulm University, Germany
{andreas.robecke,ruediger.pryss,manfred.reichert}@uni-ulm.de

Abstract. DBIScholar is a free iPhone App that allows for the retrieval and analysis of academic citations. As raw input DBIScholar uses data from Google Scholar. Based on their analysis, a number of citation metrics (e.g., h- and g-index, total number of citations) is calculated. Result are available on screen, but can be also stored and used by other Apps (e.g., email). We believe that DBIScholar and its services will be useful for authors to track the evolution of their citation metrics. Future releases will also cover other mobile platforms (e.g., Android).

Keywords: Citation Metrics, iPhone App, H-Index, G-Index

1 Introduction

Smart phones have become an indispensable work equipment for many people. As technology is evolving, CPUs have become faster, memory larger and batteries more efficient. Techniques like UMTS as well as the increasing coverage of WLAN access points provide fast mobile Internet connections and enable mobile client applications to communicate with servers located anywhere. Being equipped with a GPS unit together with a respective framework further allows for localisation and navigation functionalities within mobile applications. In the meantime smart phones have proven to be a useful platform for everyday applications. However, it still has to be proven whether contemporary smart phone technology is ready for enabling more sophisticated business applications. We evaluated this by realizing advanced applications on the iPhone as one of the most advanced smart phone platforms currently available. An important issue was to explore the capabilities and restrictions existing for iPhone application development. To elaborate on this we developed the DBIScholar iPhone App for calculating citation indices based on Google Scholar data. Since the application turned out to be more mature than a prototype and also provides interesting features for the scientific community we have decided to make it available for free. It can be downloaded from Apple's App Store [4].

The major DBIScholar feature presented in this paper is to calculate two scholarly indices, namely the *h-* and *g-index*. Both indices aim at measuring the productivity and impact of the work published by a scholar. They are based on the number the top most cited publications are referenced by other papers.

The h-index was defined by J. E. Hirsch in his paper "An index to quantify an individual's scientific research output" [11] in 2005 as the maximum number n of papers with citation numbers $>= n$. The g-index, in turn, was introduced by Leo Egghe in his paper "Theory and practice of the g-index" [9] in 2006 as an improvement to the h-index. The g-index is defined as the unique number such that the n most cited articles (together) received at least n^2 citations. The definition of the g-index inherits most properties of the formula of the h-index. In addition, it better takes into account the (few) very best cited articles of an author. The h-index is robust in the sense that it is insensitive to lowly cited papers as well as outstanding highly cited papers. Egghe claims the latter to be a drawback as the evolution of the most cited papers is not being taken into account at all. Once a paper is selected to belong to the top h papers, it does not influence the calculation of the h-index in subsequent years even if it doubles its number of citations [13]. However, the h-index still seems to be the most popular metrics used for citation analysis.

There are online resources listing the h-indices of the best scholars in their field such as the website "The h Index for Computer Science" [14] and "Arnetminer" [1]. To calculate the h-index of a scholar accurately, it is necessary to know the exact citation counts of her publications. There are a few comprehensive data sources such Web of Science [7], Scopus [6] and Google Scholar [2], which can provide this data in a more or less accurate manner. [8] and [10] discuss and compare these sources in the context of calculating h- and g-indices. Calculation results definitely vary depending on the data source used, and different opinions exist which data source serves best for such calculations. Due to the fact that Google Scholar is the only data source freely available, we use it for our application. As an advantage search results can be reproduced by anyone.

The remainder of this paper is organized as follows. Section 2 gives a short overview of the DBIScholar architecture. Section 3 describes its main features in detail. Finally, Section 4 gives an outlook on future work.

2 DBIScholar Architecture

As aforementioned our original goal was to analyse smart phone technologies in terms of business capabilities. Therefore, we considered the idea to develop an application for calculating scholarly indices to be an appropriate task as it involves data retrieval, processing, storage, and visualization. To study as many aspects of iPhone application development as possible and to gain experience on the performance and capabilities of contemporary devices, we decided to implement all functionality within the application instead of outsourcing parts to a server. The flow chart from Fig. 1 illustrates the workflow comprising data retrieval, processing and storage in DBIScholar.

To calculate the h- and g-indices of a particular author, the application's initial interface presents a form to enter the scholar of interest. An advanced form additionally allows the user to specify further details such as publication dates and subject areas similar to the advanced search form offered by Google

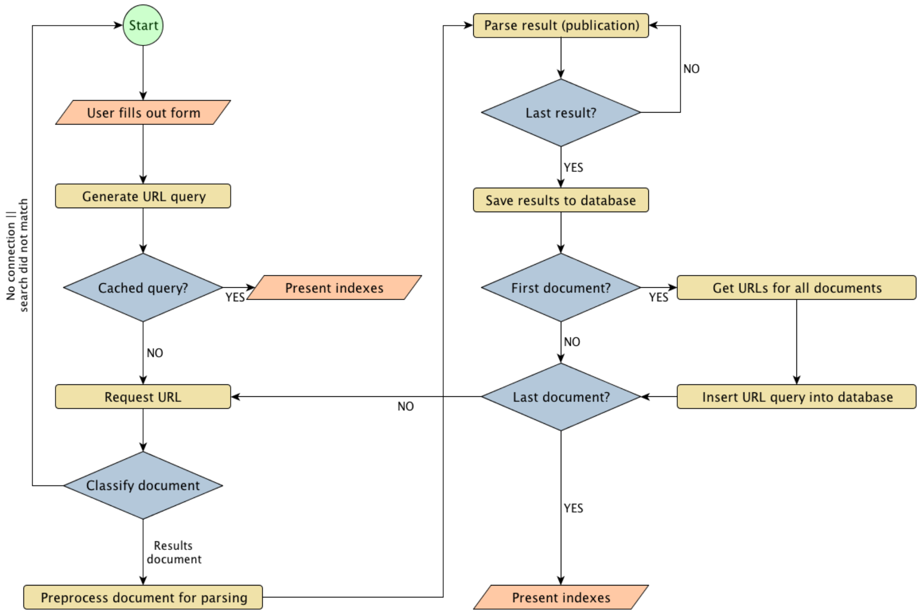


Fig. 1: DBIScholar Search Workflow

Scholar [3] and tools like Publish & Perish [5]. After specifying an author and submitting the filled search form, a respective URL is created to request the corresponding document containing the data of interest from Google Scholar. In the given context this data includes all citation counts concerning the publications of the specified scholar. If there is no cached data, the application requests the document from the created URL. Otherwise, the indices are calculated based on cached data and are immediately presented to the user. If the search returns a valid document it is pre-processed for parsing. In this context we had to consider that any document provided by Google Scholar may contain a maximum of one hundred results (publications). If a scholar has published a higher number of papers, it becomes necessary to request further documents after having parsed the first one. Additionally, to implement the caching mechanism, the query URL has to be saved after parsing the first document.

3 DBIScholar Features

Index Calculation. DBIScholar features are structured into three tabs. The left tab is the one initially presented to the user after having launched the application. Its root view presents a form allowing the user to search for the publications of a specific scholar (cf. Fig. 2a). If the search matches any result the publication counts of all retrieved publications are determined, the h- and g-indices are calculated, and the obtained results are presented to the user (cf.

Fig. 2b). The user then has the options to save the results, to email them (including all publications the results are based on), or to display the publications.

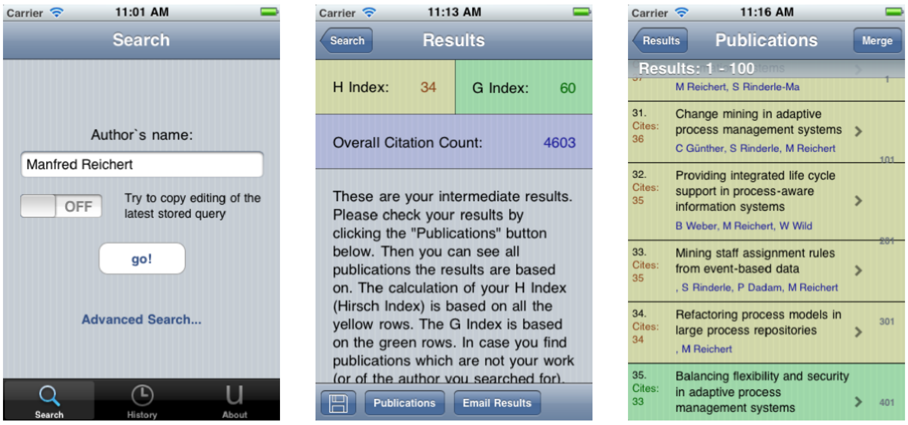
Deleting and Merging Publication Entries. To be able to calculate the indices of a scholar as precise as possible, it is necessary to check whether all publications retrieved during the search actually belong to the scholar of interest. In this context all papers actually contributing to the h- and g-indices are marked in yellow and green colours (cf. Fig. 2c). Thus it is essential to ensure that all these publications belong to the scholar of interest and not to another one with similar name or search criteria.

To allow users to discard publications of "wrong" authors, DBIScholar provides the functionality to delete publications from the list by swiping the finger across it an pressing the appearing "Delete" button. Deleted publications are added at the end of the list to the "Rejects" section. Deleting publications from the "Rejects" section, in turn, allows restoring them. DBIScholar assumes that publications can be uniquely identified by their titles. Occasionally, it happens that Google's search engine delivers multiple publications with the exact same title as result of a search. It may also happen that the same publication is listed multiple times because it was published under different titles or in different form (e.g. as journal paper and a technical report). All cases in which the same publication is listed multiple times are undesirable and are most likely caused by "confusion" of the search engine. In such a case the items should be merged into a single representative publication, and all citation counts be added up. Generally, it is up to the user to figure out which publications are the same. DBIScholar then allows her to merge the identified publications using the "Merge" button in the navigation bar of the *publications list view* (cf. Fig. 2c). In DBIScholar it is possible to merge any publication with others. The publication with the highest citation count is then kept as the representative publication and is marked with "Merge #count" in red colour in the publications list view (where count corresponds to the number of papers merged with this publication).

Advanced Search Form. In the initially presented root view (cf. Fig. 2a), the user has the option to extend the form by tapping "Advanced Search...". Amongst others, the advanced search form allows users to specify a subject area or to limit the publications to a certain period of time. This can be especially useful when search results include many publications of an author or multiple authors with similar names. Instead of deleting all the publications of "wrong" authors, it can be an option to use the advanced search to retrieve less non-fitting publications.

Displaying Publications. After selecting any publication in the list view, DBIScholar provides further details on it. The appearing user interface offers up to four additional functions. First, if a link to the publication file is available the respective document can be displayed. Second, an "Email Link" button allows users to send this link via email. Third, if the publication has been merged with others, the ”Dissolve” button displays a user interface which allows demerging existing items. Fourth, statistics on citations of the respective paper can be displayed (cf. Sect. 3).

Reuse Settings of a Previous Search. Both the search form and the advanced search form allow users to re-apply the settings they have made in the context of a previous query (e.g., to merge certain publications or to delete them). For this purpose DBIScholar allows them to turn on this feature before submitting a query. The App then checks whether the user has submitted the same query before. In this case, it will look for all settings the user has performed on the publications list of her previous search. These are then applied to the publications list obtained as result of the current query as well. However, there are cases for which this feature does not perform well. Occasionally, a single query delivers multiple publications with the exact same title. As another exception a search request might not deliver exactly the same title for an identical publication as another search. Actually, this happens in rare cases. DBIScholar then is unable to figure out that the two publications are actually the same.



Author`s name: Manfred Reichert

OFF Try to copy editing of the latest stored query

H Index: 34 G Index: 60

Overall Citation Count: 4603

These are your intermediate results. Please check your results by clicking the "Publications" button below. Then you can see all publications the results are based on. The calculation of your H Index (Hirsch Index) is based on all the yellow rows. The G Index is based on the green rows. In case you find publications which are not your work (or of the author you searched for),

Results: 1 - 100

M Reichert, S Rinderle-Ma

31. Cites: 36 Change mining in adaptive process management systems
C Günther, S Rinderle, M Reichert

32. Cites: 35 Providing integrated life cycle support in process-aware information systems
B Weber, M Reichert, W Wild

33. Cites: 35 Mining staff assignment rules from event-based data
, S Rinderle, P Dadam, M Reichert

34. Cites: 34 Refactoring process models in large process repositories
, M Reichert

35. Cites: 33 Balancing flexibility and security in adaptive process management systems

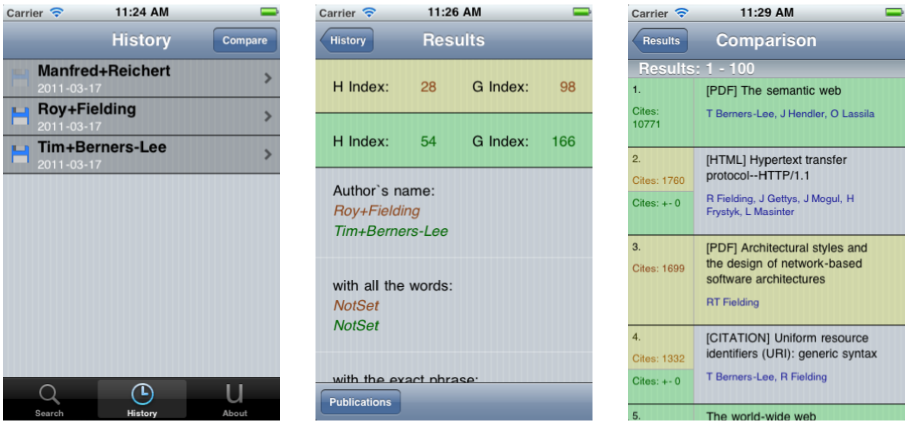
(a) Search (b) Search Results (c) Publications

Fig. 2: Search Tab

Managing Search Results. The history of all conducted and stored queries may be displayed by tapping the ”History” tab on the root view of DBIScholar (cf. Fig. 3a). The history list of previous queries then differentiates between cached and saved queries indicated either by a transparent or a shiny blue floppy disk (cf. Fig. 3a). A history entry itself displays the author’s name and the date of the query. When selecting an entry from the history list, the displayed user interface is exactly the same as the one obtained when initially triggering a search. Thus all features introduced so far are also accessible within the "History" tab. The only difference is that the results view, which displays the calculated indices, provides further details of the queries instead of instructions on how to edit the results after initially triggering a query.

Comparison. The root view of the history tab provides another interesting feature offered by the "Compare" button in the navigation bar (cf. Fig. 3a). After pressing this button the user may select two queries form the history to compare their results with each other. Doing so the two indices for both queries are displayed in the same screen (cf. Fig. 3b). The two colours yellow and green, which are also used to differentiate between publications solely contributing to the h-index and those contributing to the g-index, are now being used to differentiate between the two results (cf. Fig. 3b). Again, it is possible to display the publications the results are based on. The publications of both searches are then displayed in the same list (cf. Fig. 3c). The colours indicate to which of the two searches a publication belongs. If one publication belongs to both search results, it is displayed in neutral colour (i.e. blue). To compare the citation counts of the two searches anyway, these are displayed in a small section on the left. The citation count of the publication belonging to the second search (green) is being displayed in relation to the first one and thus may display values such as +1, −3, or 0. Note that this feature is not only interesting to compare two scholars, but also to track changes of the citation counts of publications belonging to the same author and their impact on the indices over time. This can be done by submitting the same search once in a while. When comparing the latest search with a previous one, users can find out whether citation counts of their papers have changed or whether this led to updated indices.

Graph Feature. DBIScholar allows users to visualize the evolution of the citation counts of a particular publication over time. The respective graph is based on all publications citing this publication. Particularly, it enables predictions on the future evolution of the citation counts. Users have the option to switch between a detailed view (cf. Fig. 4a) providing the exact citation counts per year, and another progression graph (cf. Fig. 4b) showing the progression of the citation counts on a single screen. Since Google Scholar does not always provide a publication date, accuracy of the graphs varies. A percentage icon on top of the graph view indicates (cf. Fig. 4a) its accuracy in percentage based on the total publication count allocated in the graph. By clicking the percentage icon, this exact issue is described in a separate screen as illustrated by Fig. 4c. Finally, it is also possible to send an image of the graph to any email address.



Manfred+Reichert
2011-03-17

Roy+Fielding
2011-03-17

Tim+Berners-Lee
2011-03-17

H Index: 28 G Index: 98

H Index: 54 G Index: 166

Author`s name:
Roy+Fielding
Tim+Berners-Lee

with all the words:
NotSet
NotSet

with the exact phrase:

Results: 1 - 100

1. Cites: 10771 — [PDF] The semantic web — T Berners-Lee, J Hendler, O Lassila

2. Cites: 1760 Cites: +- 0 — [HTML] Hypertext transfer protocol--HTTP/1.1 — R Fielding, J Gettys, J Mogul, H Frystyk, L Masinter

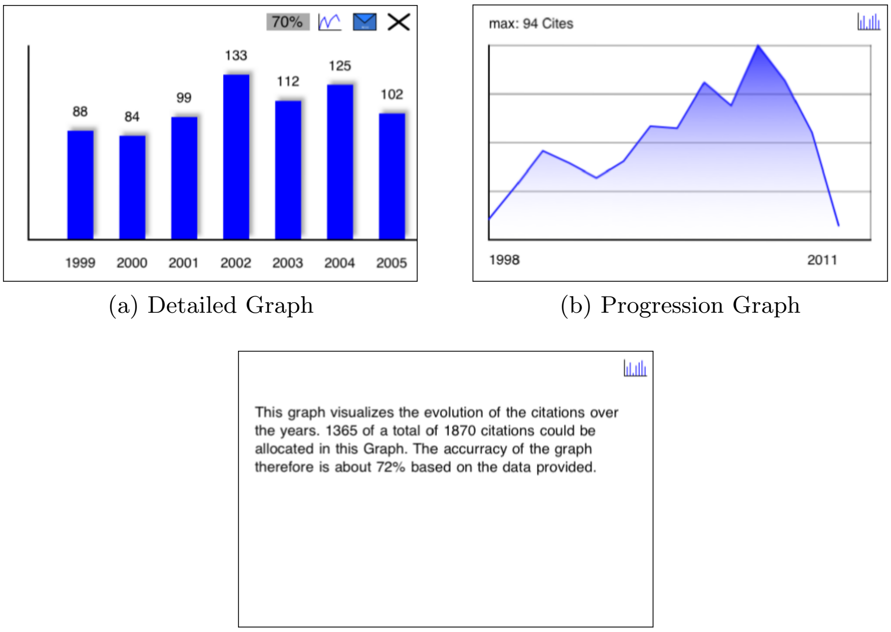
3. Cites: 1699 — [PDF] Architectural styles and the design of network-based software architectures — RT Fielding

4. Cites: 1332 Cites: +- 0 — [CITATION] Uniform resource identifiers (URI): generic syntax — T Berners-Lee, R Fielding

5. The world-wide web

(a) History Root View (b) Index Comparison (c) Citation Comparison

Fig. 3: History Tab



(a) Detailed Graph (b) Progression Graph

This graph visualizes the evolution of the citations over the years. 1365 of a total of 1870 citations could be allocated in this Graph. The accurracy of the graph therefore is about 72% based on the data provided.

(c) Graph Accuracy

Fig. 4: Graph Views

4 Outlook

Since it turned out to be a mature application providing useful features for researchers, we will invest further efforts to enhance DBIScholar. One idea is to visualize the evolution of the indices based on the dates of all publications citing an author of interest. This feature would be analogue to the one described in Section 3 for the citation counts of one particular paper. Another extension will be the integration of an offline access feature allowing users to download publications to access them at any time. Some articles about the h index claim that it is possible for authors to manipulate their indices by citing their own publications [12]. Thus, another interesting feature is to count the number of self-citations and to calculate the percentage based on the overall citation count. To further improve DBIScholar and to extend it with additional features we would appreciate getting feedback via Apple's App Store rating system [4] or email. Finally, we will provide DBIScholar on other mobile platforms (e.g. Android and Windows Phone 7) in future.

References

1. Arnetminer. URL `http://www.arnetminer.org/`.
2. Google scholar. URL `http://scholar.google.com/`.
3. Google scholar - advanced scholar search. URL `http://scholar.google.de/advanced_scholar_search?hl=en`.
4. itunes dbischolar app store link. URL `http://itunes.apple.com/us/app/dbisscholar/id401979619?mt=8&ls=1`.
5. Publish or perish software. URL `http://www.harzing.com/pop.htm`.
6. Scopus. URL `http://www.scopus.com/`.
7. Web of science. URL `http://isiwebofknowledge.com/`.
8. Judit Bar-Ilan. Which h-index? – a comparison of wos, scopus and google scholar. *Scientometrics, Vol. 74, No. 2 (2008) 257–271*, 2007.
9. Leo Egghe. Theory and practise of the g-index. *Scientometrics*, 2006.
10. Prof. Anne-Wil Harzing. Google scholar - a new data source for citation analysis. 2007.
11. J E Hirsch. An index to quantify an individual's scientific output. *Proc. Natl. Acad. Sci. U. S. A.*, 2005.
12. Konstantin V. Krutovsky Lev A. Zhivotovsky. Self-citation can inflate h -index. *Scientometrics*, 2008.
13. Lokman I. Meho and Kiduk Yang. A new era in citation and bibliometric analyses: Web of science, scopus, and google scholar. *Journal of the American Society for Information Science and Technology*, 2006.
14. Jens Palsberg. The h index for computer science. URL `http://www.cs.ucla.edu/~palsberg/h-number.html`.
15. Andreas Robecke. Development of an iphone business application - diploma thesis, university of ulm. 2011.

A Flexible System for Ontology Matching

Ngo DuyHoa, Zohra Bellahsene, Remi Coletta

LIRMM, Univ. Montpellier 2
34392 Montpellier, France
firstname.name@lirmm.fr

Abstract. Most of solutions provided by current ontology matching tools lack flexibility and extensibility namely for adding new matchers and dealing with users' requirements. In this paper, we present a system YAM++, which supports self-configuration, flexibility and extensibility in combining individual matchers. Moreover, it is more human-centered approach since it allows users to express their preference between precision and recall. A set of experiments over OAEI benchmark dataset demonstrate its effectiveness and efficiency in terms of quality of matching and flexibility of the system.

Keywords: Ontology matching, data mining, flexibility, self-configuration, cost-sensitive classification.

1 Introduction

Ontology matching is needed in many application domains. For example, the possibility of content-based query of the semantic Web depends only on the capacity of the system to find correspondences (mappings) between ontologies of the related information sources. Many diverse solutions of matching have been proposed so far; however, there is no integrated solution that is a clear success, which is robust enough to and flexible be the basis for future development, and which is usable by non expert users.

In this paper, we present our system YAM++, which supports self-configuration, flexibility and extensibility in combining individual matchers. To demonstrate the important of the flexibility in terms of system extensibility and user preference, let's us introduce two scenarios that frequently arise when people study schema and ontology matching.

In the first scenario, reseachers and developers of a matching system usually have to supplement new invented similarity metrics or update existing metrics with the new ones. According to [10], similarity metrics are also known as individual matchers. In both situations, developers must estimate the degree of contribution of these metrics and then find a suitable model to combine them. The estimation and combination models are normally tested carefully on existing "gold standard" datasets first, before applying them to real scenarios. In that case, a flexible system will help them to automatically deal with new metrics.

In the second scenario, imagine that users run a matching system to find all mapping pairs of entities between two ontologies. A matching system, generally, outputs a list of candidate mappings and corresponding confident values. Users then must verify these mappings in order to remove incorrect ones. This process will not take much time because number of the suspect mappings is limited. Next, users need to find missing mappings which matching system did not discover. This process is very time consuming because it will be done manually on a huge number of candidate mappings. The manual effort of this phase is called post-match effort. Users may spend many hours or even few days to finish this work. Therefore, users desire to have a way to improve number of correct mappings in order to reduce post-match effort.

Based on these scenarios, the motivation of our system can be described as follows: Giving two ontologies represented in some ontology languages (N3, RDF, OWL, etc.), find a flexible approach to combine individual matchers with the following features: (i) achieving high matching quality result (*precision, recall and f-measure*), (ii) system's self-configuration, (iii) system's extensibility, (iv) generating a dedicated matcher according to the user's preference between precision and recall.

The remainder of this paper is organized as follows: In section 2, we describe our ontology matching system in detail. In Section 3, we present the results of experiments performed to highlight the main interesting features of our ontology matching tool. Section 4 contains the related work. Finally, Section 5 contains concluding remark about our system.

2 YAM++ Ontology Matching System

Our approach has been implemented in YAM++ - (not) Yet Another Matcher system for ontology matching. It follows the same approach used in YAM schema matching system [2]. However, the YAM++ aims to work with ontology matching, which is semantically richer than XML schema. For this purpose, we added new features such as:

- New similarity metrics working with different features (e.g. name, label, comments, relations) of ontologies' entities.
- New dictionary metrics based on different algorithms.
- New metrics based on information retrieval technique calculate similarity score between context and descriptive information of entities.
- Graphical user interface for setting parameters, displaying and verifying discovered mappings returned from system.

The main components of YAM++ system are depicted in Figure 1. It only requires as input, the set of ontologies to be matched. However, the user can also provide additional inputs, i.e., some preferences between precision and recall.

The **Knowledge Base** is a system repository, containing library of similarity metrics and library of learning models. It also stores list of gold standard datasets, which is a pair of ontologies with expert mappings between some of their entities built by domain experts.

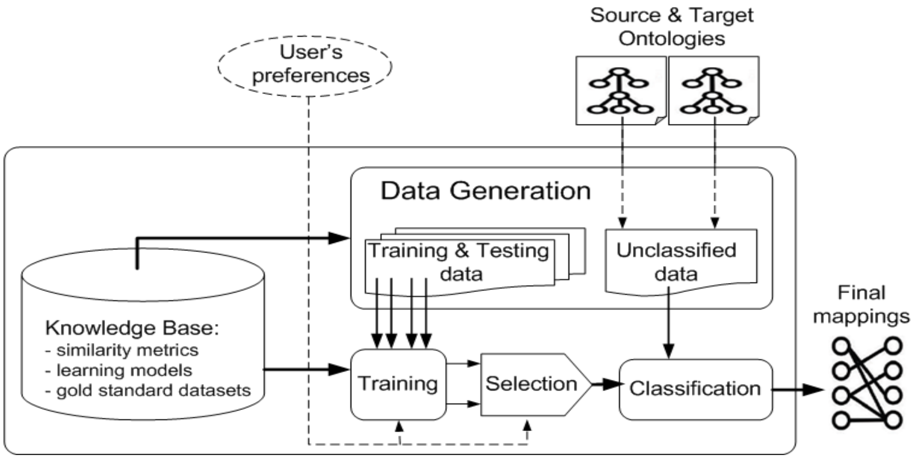


Fig. 1. YAM++ system architecture

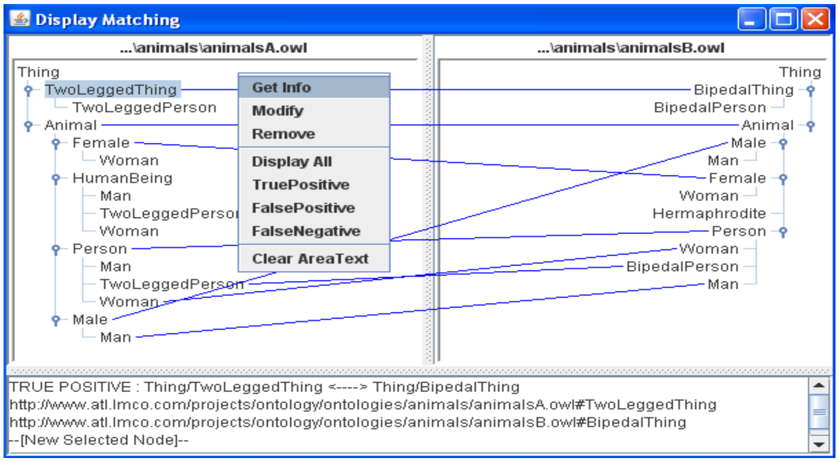
The **Data Generation** module transforms gold standard datasets and input ontologies to learning (including training and testing) and unclassified data respectively. The idea is that each pair of entities (e_i, e_j) becomes a machine learning instance, which its features are the similarity score calculated by similarity metrics on (e_i, e_j). In training and testing data, the class of instances is determined by the confidence value of the corresponding pair (e_i, e_j) in the expert mappings set. For unclassified data, the class of instances is set to unknown value.

The **Training** module finds the optimal configuration for each learning model according to training data passed from **Data Generation** module. Besides, it can also take a user preference for either Precision or Recall in training process to generate classification models that favor this preference. The configuration process is automatic and transparent to users. The average performances (Precision, Recall, F-Measure) of all learning models achieved from running 10-fold cross validation and different testing data are temporarily saved for comparison purpose.

The **Selection** module by default will select a classification model, whose the obtained average F-Measure is highest. If user provides a preference between Precision and Recall, a classification model, which obtains the best result corresponding to this preference, is selected for next stage. In this paper, we call it **dedicated matcher** or **dedicated model.**

In the **Classification** module, a dedicated matcher predicts each instance in unclassified data by a predicted value. If the classification model is nominal, the predicted value is TRUE or FALSE, which means two entities corresponding with classified instance are matched or not.

Finally, these mappings are displayed in graphical user interface. Users can judge a mapping whether it is correct or not by their knowledge of ontologies' domain. Users also can modify, remove incorrect mappings or add new mappings with the help of command operations appeared in system's menu (see Figure 2).



```
TRUE POSITIVE : Thing/TwoLeggedThing <----> Thing/BipedalThing
http://www.atl.lmco.com/projects/ontology/ontologies/animals/animalsA.owl#TwoLeggedThing
http://www.atl.lmco.com/projects/ontology/ontologies/animals/animalsB.owl#BipedalThing
--[New Selected Node]--
```

Fig. 2. User interface for mappings visualization

3 Experiments

In this section, we present the capabilities of our system using two experiments:

1. We show the flexibility and extensibility of our system in term of integrating new similarity metrics automatically and transparently to users.
2. We show another ability of generating a dedicated matcher based on the user's preference (promoting recall).

3.1 Experiment 1: Flexibility and Extensibility

For demonstration purpose, we run two scenarios and compare their performance quality.

- In the first scenario, we use a set of string metrics as individual matchers to calculate similarity value between entities based on entities' names and labels. These metrics are taken from open source code library SecondString[1], SimMetric[2]. These metrics are Levenstein, SmithWaterman, JaroWikler, Stoilos, QgramDistance, MongeEklan and Level2 metrics.

[1] http://secondstring.sourceforge.net/
[2] http://sourceforge.net/projects/simmetrics/

- In the second scenario, we add metrics working with dictionary WordNet[3] to exploit semantic features and metrics working with entities' description. We have implemented Lin and WuPalmer[7] algorithm for dictionary metrics. For comparing descriptive information, we construct a text corpus for each entity. Entity's corpus consists of its meta-data (name, labels, comments), meta-data of its related entities (sub-concepts, sub-properties, restricted properties, range). A Vector Space Model is constructed from these copora [8]. By using TF*IDF algorithm for term weighting, each corpus is transformed to a feature vector. The similarity score of two entities is calculated by cosine similarity of their feature vectors.
- In both scenarios we train different learning models such as: tree-based (J48, CART, ADTree, NBTree), probability-based (NaiveBayes, BayesNet), function-based (SMO, LibSVM, Logistic, MultiLayerPerceptron), instance-based (IBk, NNge, VFI). These models are taken from open source Weka[4] library. The gold standard datasets are taken from OAEI[5] and I3CON[6] repositories. In both scenarios, we do not set preference between Precision and Recall, so the criterion for selection is maximum F-Measure. The winner model after running selection process in both cases is DecisionTree J48 model. It means that the dedicated models used in Classification module are a trained J48 in both the scenarios.
- For comparison purpose, we run matching on set of datasets of OAEI 2009: {**#104, #203, #204, #205, #206, #201, #201-2, #201-4, #201-6, #201-8**}.

The observations from Figure 3 are:

- On datasets **#104, #203 and #204**, both scenarios have the same F-Measure (≈ 1.0). This is because the linguistic information of these test ontologies is highly similar to with that of the reference ontology. In fact, entities' names of these ontologies are identical or are modified by some naming conventions. Therefore, adding the new metrics do not have a significant impact on the result.
- On datasets **#205 and #206** entities' names in test ontologies are replaced by synonym words or are translated in another language. Thank to using metrics based on dictionary and descriptive information, the achieved average F-Measure is increased **57%** from the first (**0.36**) to the second scenario (**0.93**).
- On datasets **#201, #201-2, #201-4, #201-6 and #201-8**, entities' names in test ontologies are replaced by random sequence symbols with **100%, 20%, 40%, 60% and 80%** respectively. However, entities are also described by labels and comments, so the achieved average F-Measure is increased **45%** from the first (**0.52**) to the second scenario (**0.97**).

[3] http://wordnet.princeton.edu

[4] http://www.cs.waikato.ac.nz/ml/weka/

[5] http://oaei.ontologymatching.org/2009/

[6] http://www.atl.lmco.com/projects/ontology/i3con.html

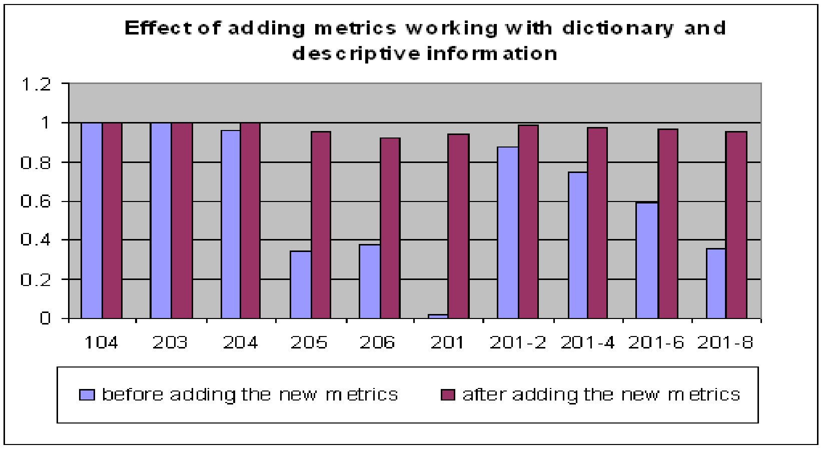


Fig. 3. Comparison on F-Measure of all datasets in the first and the second scenarios

The most interesting feature to note is that the process of reconfiguration system with new metrics is totally automatic and transparent to the users.

3.2 Experiment 2: Promoting Recall

Traditionally, the measure used to compute performance quality of matching tools, is the F-Measure: a combination of Precision (the ratio of correctly found correspondences (a.k.a true positive) over the total number of returned correspondences [5]) and Recall (the ratio of correctly found correspondences over the total number of expected correspondences [5]), in which precision and recall have the same weight. F-Measure makes sense when using matching tool as black box, without any user validation. But, most of the time, the user have to perform some post-match effort in order to discard some irrelevant and discover the missing mappings. In this experiment, we demonstrate the impact of user preference between Precision and Recall on post-match effort.

Technically, most classification models suffer from two errors during classifying: i) discovering an irrelevant correspondence (a.k.a. false positive) and ii) missing a relevant correspondence (a.k.a. false negative). The first error decreases precision while the second one decreases recall. In order to get better result in term of recall, we need to set the cost on false negative error higher than that on false positive error. This is a well-known issue called Cost-Sensitive Learning in Data Mining [4].

In order to deal with cost-sensitive learning, we use MetaCost and CostSensitiveClassification algorithms [11]. These algorithms belong to meta-learner class. They make a wrapper on base learning models in such a way that learning models effectively minimize cost . The preference between Precision and Recall is expressed by a proportion of the cost on false negative and the cost on false positive.

We perform our experiments with different proportion values, on the real datasets in OAEI 2009: {#**301,** #**302,** #**303,** #**304**}. The base learning models, the list of similarity metrics and training data are the same as described in the second scenario in the first experiment.

	proportion = 1	proportion = 5	proportion = 10	proportion $\geq$ 15
Total True Positive	122	127	**133**	**133**
Total False Positive	40	61	**83**	97
Total Undiscovered	19	14	**8**	**8**

Table 1. The effect of promoting Recall

For each proportion value, one dedicated matcher is generated. The observations from Table 1 are:

- By increasing the proportion value, the total of candidate mappings discovered as True Positive is increased. This advantage helps users to reduce time for discovering missing mappings.
- When the proportion is equal to 10, the total number of true positive mappings is maximum. After that, only the total number of false positive increases. This is a disadvantage, because users must to remove more irrelevant mappings.
- Notice that whatever the value we set for proportion, it always remains some matches we are not able to discover automatically.

As an example, when proportion is set to 10, the dedicated matcher discovers **11** (133 - 122) additional true positives, but **43** (83 - 40) additional false positives in comparison with the default matcher. In fact, the effort for manually removing an incorrect mapping is much less than the one for discovering a new correct mapping among **9949** pairs (total candidate mappings of 4 datasets). Therefore, by promoting recall, our system reduces user's post-match effort during the validation phase.

4 Related work

There are many studies on Ontology Matching [6],[5]. In this section, we only mention the closest ones that are based on machine learning approaches.

GLUE [1] is a well-known of learning-based ontology mapping system. GLUE uses a set of base learners to exploit different type of information from instances and taxonomy structures. Then, it uses a meta-learner to combine these base learners to achieve higher classification accuracy than any single base learner alone. The drawback of GLUE is that it requires a large number of instances associated with the nodes in taxonomies, whereas most ontologies do not contain these information. YAM++ is different with GLUE in that YAM++ uses machine learning approach to combine different individual matchers which exploit different features of entities such as name, description and structure information. YAM++ does not exploit information of instance associated with entities.

Another systems using machine learning approach for ontology mapping such as APFEL [3] and [9]. Our approach and these systems are quite similar in the way of using machine learning approach to combine different similarity metrics. However, in YAM++, we use some other data mining techniques to help users reduce the post-match effort.

5 Conclusion

In this paper, we present a flexible system for ontology matching task that proves the following interesting features:

- Flexibility and extensibility in terms of combining individual matchers.
- Generating a dedicated matcher according to the user's preference.

We have developed a prototype which has been tested with the datasets of OAEI 2009 benchmark. Through these experiments, we have validated the features listed above.

References

1. AnHai Doan, Jayant Madhavan, Pedro Domingos, and Alon Y. Halevy. Ontology matching: A machine learning approach. In *Handbook on Ontologies*, pages 385–404. 2004.
2. Fabien Duchateau, Remi Coletta, Zohra Bellahsene, and Renée J. Miller. Yam: a schema matcher factory. In *CIKM*, pages 2079–2080, 2009.
3. Marc Ehrig, Steffen Staab, and York Sure. Bootstrapping ontology alignment methods with apfel. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, WWW '05, pages 1148–1149, New York, NY, USA, 2005. ACM.
4. Charles Elkan. The foundations of cost-sensitive learning. In *In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, pages 973–978, 2001.
5. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
6. Yannis Kalfoglou and W. Marco Schorlemmer. Ontology mapping: The state of the art. In *Semantic Interoperability and Integration*, 2005.
7. Feiyu Lin and Kurt Sandkuhl. A survey of exploiting wordnet in ontology matching. In *IFIP AI*, pages 341–350, 2008.
8. Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, 2008.
9. Ming Mao, Yefei Peng, and Michael Spring. Ontology mapping: As a binary classification problem. *Semantics, Knowledge and Grid, International Conference on*, 0:20–25, 2008.
10. Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.
11. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, October 1999.

SD Elements: A Tool for Secure Application Development Management

Golnaz Elahi[1], Tom Aratyn[2], Ramanan Sivaranjan[2], Rohit Sethi[2], and Eric Yu[3]

[1] Department of Computer Science, University of Toronto, Canada, M5S 1A4
gelahi@cs.toronto.edu
[2] SD Elements, Toronto, Canada
Tom,Ramanan,Rohit@sdelements.com
[3] Faculty of Information, University of Toronto, Canada, M5S 3G6
eric.yu@utoronto.ca

Abstract. A major problem in achieving security goals in application development is the overwhelming amount of security-related information, variety of tools, and numerous security risks and vulnerabilities. Software analysts, developers, and testers are not often able to identify relevant security knowledge. Many security tools focus only on detecting vulnerabilities, but the embedded available security guidelines are usually not directly auditable. To fill these gaps, we introduce a new tool, called SD Elements, which focuses on prevention of vulnerabilities as opposed to detection. SD Elements is a centralized security knowledge base that covers different development life cycle phases, so security is built into the application from the early phases of the life cycle. Users are able to specify technologies, platforms, requirements, and programming languages, and SD Elements tailor security guidelines for different projects according to the user specifications. It enables businesses to provide tangible security audit evidence and trace compliance with security standards. The tool is currently being beta tested in varieties of firms, by different roles, and in different development phases.

Keywords: Application security, security requirements, development guidelines, security knowledge, test case.

1 Introduction

The software engineering community is slowly beginning to realize that information security is also important for software whose primary function is not related to security [1]. Since prevention is often more economical than remediation, empirical security knowledge such as common attacks and vulnerabilities are made public and available for practitioners through web-based portals such as NVD [2], CWE [3], OWASP [4]. Security standards, such as PCI DSS [5] or ISO [6] provide high level guidelines and impose several compliance requirements to application developers.

In reality, software analysts, developers, and testers are overwhelmed with the amount of available information and variety of tools they can employ. Analysts are given massive lists of security requirements, guidelines, and standards, and they need to provide tangible and audit-able evidence that the products comply with security guidelines. Employing tools can help application testers. However, security testing tools are usually intimidating and their adoption rate is low, specially because application developers and testers are not often security experts. Testers also need to tailor the testing scripts for the platforms and technologies that they use.

The bottom line for practitioners is finding the relevant body of information to their projects. However, there is a significant gap between the existing body of empirical knowledge collected in (web-based) knowledge portals and actual development demands.

1.1 Contributions

To fill the current gaps, we introduce a secure application development management tool, called SD Elements, which provides a set of core values to application developers, system analysts, and quality assurance teams.

SD Elements is a web-based knowledge repository of security guidelines, empowered by a retrieval tool. SD Elements surveys users to learn about the nature of the project, platform, language, and technologies, and then it tailors security knowledge:

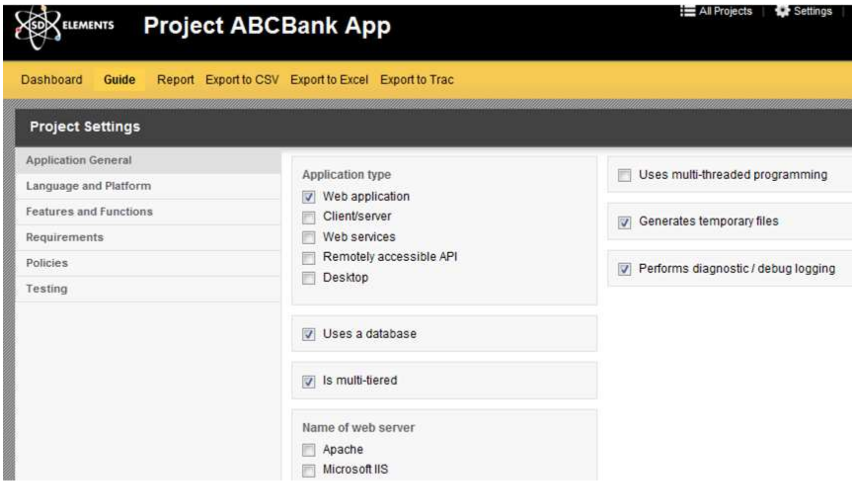
- Generates relevant security requirements.
- Provides tailored guidelines on secure architecture design.
- Provides reusable development standards for different development platforms and technologies.
- Provides sample tested code for implementing the standards.
- Creates a list of security test cases (and check lists) to enable non-expert developers systematically test security requirements.
- Integrates into application life cycle management and bug tracking tools such as Quality Center and trac.
- Ranks the risks related to standards which helps with prioritization of development guidelines and security requirements.

SD Elements focuses on vulnerability prevention instead of detection. It integrates security knowledge into the development life cycle, thus, security is built into the application from the early phases. SD Elements provides a compliance mechanism, i.e., users can trace which guidelines are employed, implemented, and tested. This provides businesses with tangible and traceable evidence for audit purposes. Finally, it provides requirements, implementation, and testing guidelines, in situations that compliance with PCI DSS and HIPPA is needed.

2 SD Elements Architecture

SD Elements users will be software developers such as requirements analysts, programmers, testers, project leaders, and security analysts. For each project, SD Elements surveys the developers about the nature of the project, security features and users of the application, types information being handled by the application, business drivers and policies, platforms, technologies, programming languages, and application interfaces. By collecting these information about the project, SD Elements retrieves a customized set of guidelines and requirements, appropriate for specific projects.

For example, application general questions (Figure 1) uncover the type of application, type of web server, programming languages, platforms, third party technologies and libraries. The survey also enables users to provide more details about the features and functions of the project. For example, developers can specify whether the application being developed involves interactions with operating system, file-upload function, authentication of end users, etc.



Application type
☑ Web application
□ Client/server
□ Web services
□ Remotely accessible API
□ Desktop

☑ Uses a database

☑ Is multi-tiered

Name of web server
□ Apache
□ Microsoft IIS

□ Uses multi-threaded programming

☑ Generates temporary files

☑ Performs diagnostic / debug logging

Fig. 1. SD Elements Survey (application general questions)

The answer to the questions enable the tool to refine the rest of questions as well as retrieve relevant security guidelines and requirements. These guidelines in the SD elements knowledge base are developed according to:

- Current vulnerabilities for different technologies, platforms, and languages. Each guideline or requirement corresponds to a vulnerability in Common Weakness Enumeration list of vulnerabilities [3].
- Best practices and existing standards such as OWASP [4], WASC threat classification [7]; empirical data about commonly-exploitable applications in web applications based on years of penetration testing; threat models, and source code review; and regulatory compliance including PCI DSS, HIPPA HITECH, GLBA, NERC CIP, and international privacy laws.

2.1 Knowledge Retrieval Engine

SD Elements' knowledge storage and retrieval is based on Boolean logic. The knowledge retrieval engine provides security guidelines, based on what users has specified. For example, if users select a J2EE project, then SD Elements concludes the user will be developing a web application that is probably multi-tiered, etc. Thus it does not require overwhelming efforts for users and project owners to describe the nature of project for receiving useful information. Project managers can get security guidelines as well, without knowing much of technical details.

2.2 SD Elements Knowledge Base Architecture

Figure 2 depicts a high level overview of the SD Elements' knowledge base architecture. The contents of the knowledge base are vulnerabilities, security standards, and implementation of the standards. By answering the survey questions, a set of properties about the project are gathered. Each property entails a set of content. The tool is implemented in Django which allows creating models to generate the data base schemas.

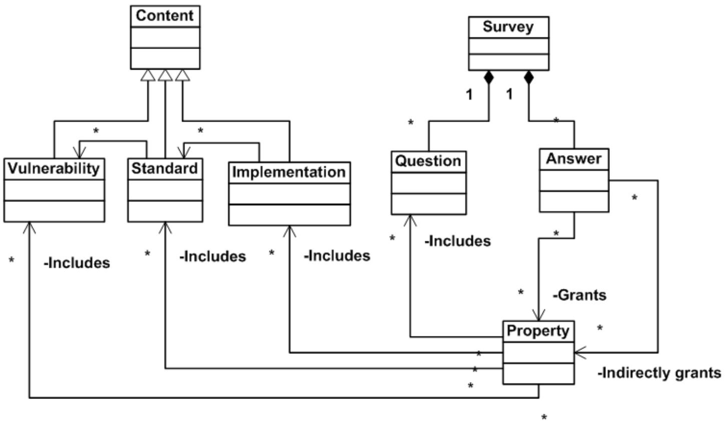


Fig. 2. High level architecture of SD Elements knowledge base

3 Integration with Software Development Life Cycle

SD Elements helps in management of secure application development. It facilitates building security into the application, from the early requirements stage, and considering security in mind at the design and implementation activities. Finally, it provides step-by-step test cases to help non-security experts test relevant cases to their application.

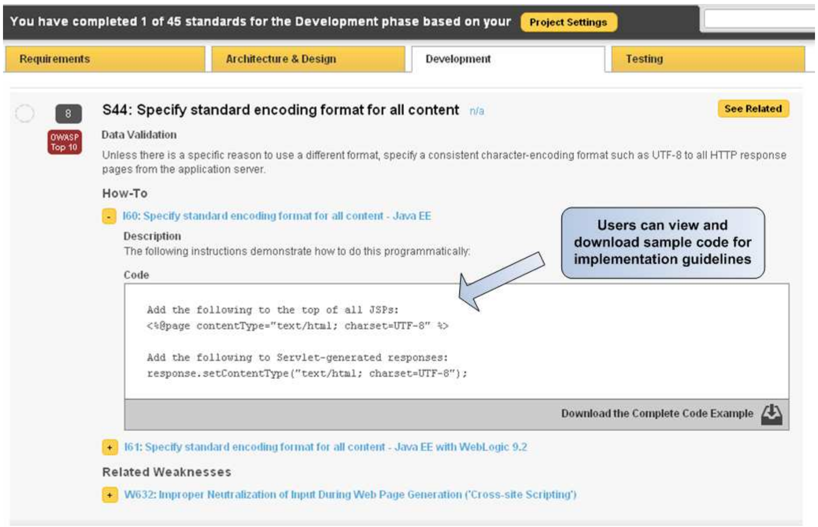
3.1 Requirements Generation

SD Elements supports application security requirements analysis:

- Surveys analysts about the project settings.
- Generates a list of relevant security requirements (in different categories) which are tailored with respect to the project settings.
- Links generated requirements to relevant vulnerabilities.
- Links the generated requirements to a design/implementation guidelines and test cases.
- Lists the requirements criteria of acceptance. These criteria are actually the test cases. Thus, analysts will know from the early stages, how the requirements will be tested.

3.2 Design and Development Guidelines

SD Elements provides project-specific implementation guidelines. Suggested guidelines correspond to the list of security requirements and settings of the project. Sample (and tested) code, whenever applicable, is provided as part of the content. For example, for the HTML encoding for JSPs, SD Elements provide a sample code as depicted in Figure 3. Figure 4 shows an example list of development guidelines in the authentication category.



You have completed 1 of 45 standards for the Development phase based on your Project Settings

S44: Specify standard encoding format for all content

Data Validation

Unless there is a specific reason to use a different format, specify a consistent character-encoding format such as UTF-8 to all HTTP response pages from the application server.

How-To

I60: Specify standard encoding format for all content - Java EE

Description

The following instructions demonstrate how to do this programmatically:

Code

```
Add the following to the top of all JSPs:
<%@page contentType="text/html; charset=UTF-8" %>

Add the following to Servlet-generated responses:
response.setContentType("text/html; charset=UTF-8");
```

Users can view and download sample code for implementation guidelines

I61: Specify standard encoding format for all content - Java EE with WebLogic 9.2

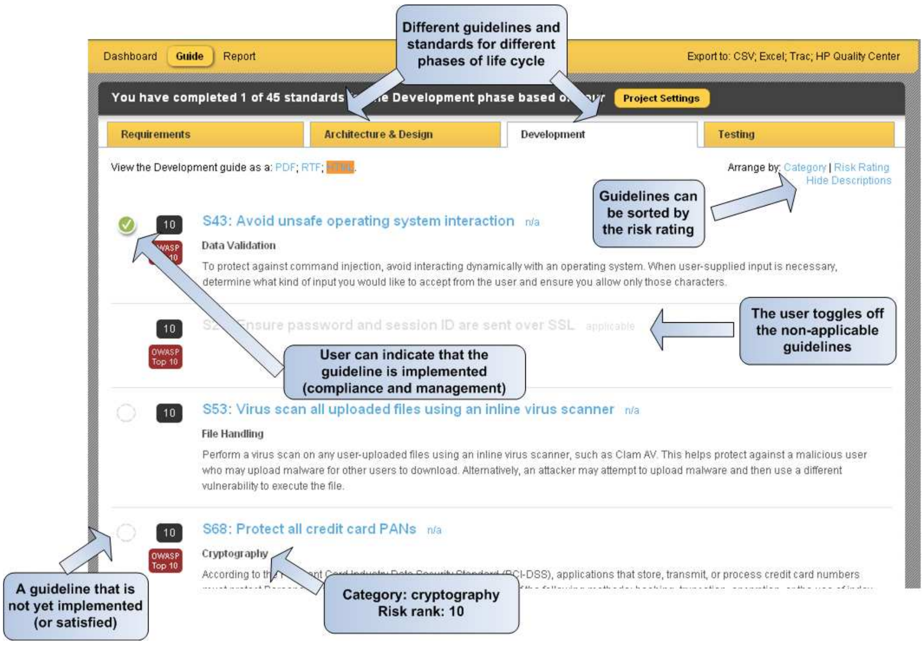
Related Weaknesses

W632: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

Fig. 3. Standard encoding format, sample code

3.3 Application Testing

SD Elements generates step by step test cases, that include failure conditions, sample scripts (if needed), guidelines about testing tools, and guiding videos. By the time the users get to the testing phase, SD Elements has provided proper security requirements and implementation guidelines. Thus, although test cases are provided, the emphasis is on preventing vulnerabilities instead of detection. Test cases are linked to security requirements, and user can specify a test is passed. Then, the requirements status is changed (to satisfied) as well. Figure 5 shows a screen shot of a sample test case.



You have completed 1 of 45 standards ... e Development phase based o... ur

S43: Avoid unsafe operating system interaction

Data Validation

To protect against command injection, avoid interacting dynamically with an operating system. When user-supplied input is necessary, determine what kind of input you would like to accept from the user and ensure you allow only those characters.

S2... Ensure password and session ID are sent over SSL

S53: Virus scan all uploaded files using an inline virus scanner

File Handling

Perform a virus scan on any user-uploaded files using an inline virus scanner, such as Clam AV. This helps protect against a malicious user who may upload malware for other users to download. Alternatively, an attacker may attempt to upload malware and then use a different vulnerability to execute the file.

S68: Protect all credit card PANs

Cryptography

According to th... ent Card Industry Data Security Standard (PCI-DSS), applications that store, transmit, or process credit card numbers

Different guidelines and standards for different phases of life cycle

Guidelines can be sorted by the risk rating

The user toggles off the non-applicable guidelines

User can indicate that the guideline is implemented (compliance and management)

A guideline that is not yet implemented (or satisfied)

Category: cryptography
Risk rank: 10

Fig. 4. A sample development guideline generated by SD Elements tool for a banking application

3.4 Umbrella Processes

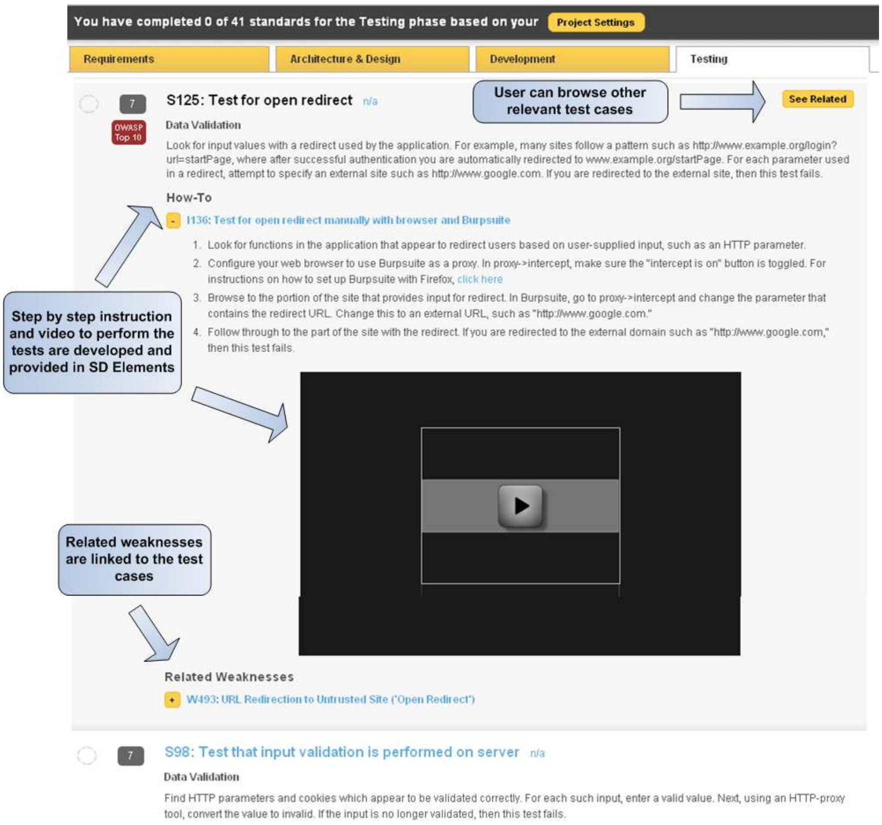
Incremental survey: The survey section of the tool can be answered gradually and incrementally, i.e., the more questions answered by the user, the more specific and accurate guidelines are retrieved by SD Elements.

A Traceability System: Users can trace which guidelines they have applied, which ones are not applicable, and which guidelines are outstanding. Applicable guidelines and implementation standards can be added to bug tracking systems and generated requirements can be imported to general requirements documents such as PDF and doc files.

4 Beta Test Plan

SD Elements will be in beta test in a variety of sectors, such as energy, independent software vendors, healthcare, and financial services. It will be mostly championed by application security managers and development team leaders. The beta tests will help us investigate various aspects of real world application of SD Elements. By the end of the beta test phase, we will have concrete data to evaluate usefulness and usability of SD Elements in real world practice.

1. How and in what logical path users browse the standards.
2. How the standards and requirements are applied in different phases of development and by what roles.


Project Settings
Requirements
Architecture & Design
Development
Testing
User can browse other relevant test cases
See Related
7
OWASP Top 10
Step by step instruction and video to perform the tests are developed and provided in SD Elements
Related weaknesses are linked to the test cases
7

You have completed 0 of 41 standards for the Testing phase based on your Project Settings

S125: Test for open redirect n/a

Data Validation

Look for input values with a redirect used by the application. For example, many sites follow a pattern such as http://www.example.org/login?url=startPage, where after successful authentication you are automatically redirected to www.example.org/startPage. For each parameter used in a redirect, attempt to specify an external site such as http://www.google.com. If you are redirected to the external site, then this test fails.

How-To

- I136: Test for open redirect manually with browser and Burpsuite

1. Look for functions in the application that appear to redirect users based on user-supplied input, such as an HTTP parameter.
2. Configure your web browser to use Burpsuite as a proxy. In proxy->intercept, make sure the "intercept is on" button is toggled. For instructions on how to set up Burpsuite with Firefox, click here
3. Browse to the portion of the site that provides input for redirect. In Burpsuite, go to proxy->intercept and change the parameter that contains the redirect URL. Change this to an external URL, such as "http://www.google.com."
4. Follow through to the part of the site with the redirect. If you are redirected to the external domain such as "http://www.google.com," then this test fails.

Related Weaknesses

+ W493: URL Redirection to Untrusted Site ('Open Redirect')

S98: Test that input validation is performed on server n/a

Data Validation

Find HTTP parameters and cookies which appear to be validated correctly. For each such input, enter a valid value. Next, using an HTTP-proxy tool, convert the value to invalid. If the input is no longer validated, then this test fails.

Fig. 5. A sample SD Elements test case

3. Whether users treat guidelines only as an after the fact check list or developers use guidelines in daily development tasks.
4. Whether SD Elements help developers actually prevent the introduction of potential vulnerabilities into the code.
5. Whether users find SD Elements intuitive and usable.
6. Whether users need additional security knowledge sources in addition to SD Elements.

5 Related Work

Various web-based software vulnerability knowledge bases provide a shared and standard way for identifying, specifying, and measuring software weaknesses and vulnerabilities. Common Weakness Enumeration (CWE) [3] provides a unified, measurable set of software weaknesses for enabling effective discussion, description, selection, and use of software security tools and services that can find these weaknesses in source code and operational systems. National Vulnerability Database (NVD) portal provides a search engine over the Common Vulnerability and Exposure (CVE) and Common Configuration Enumeration (CCE) databases. However, these knowledge portals usually lack specific guidelines to prevent the introduction of vulnerabilities. The burden of identifying relevant vulnerabilities in the massive lists of these portals is on developers. SD Elements solves these issues by providing customized guidelines for preventing CWE vulnerabilities.

The need for security guideline customization has been addressed in another tool called TeamMentor [8]. TeamMentor only provides two layers of guidelines filtering: 1) technology and 2) role of the user. Thus, still a massive list of guidelines without specific categorization is provided to the user. A link between the suggested guidelines for different phases of life cycle is not considered, thus traceability is not possible. Project specific features and functionalities are not used to generate the list of guidelines, and still software developers can become overwhelmed with the large list of guidelines.

6 Conclusions and Future Work

SD Elements is a web-based security knowledge base that provides security guidelines for different development life cycle phases. The main goals of SD Elements is to build security into the application, from the early phases of the life cycle. The outstanding contribution of SD Elements is tailoring the guidelines according to project description. SD Elements helps businesses provide tangible security audit evidence and trace compliance with security standards.

The results of the beta tests will help us investigate whether by applying SD Elements, more vulnerabilities are actually prevented. In future releases, SD Elements will be customizable to different domains and businesses. End user administrators will be able to add their own questions, answers, and content items so that they can support any technology stack. Also, we will continuously add more content, thus SD Elements will work as a subscription service rather than a single tool.

References

1. I. A. Tondel, M. G. Jaatun, and P. H. Meland, “Security requirements for the rest of us: A survey,” *IEEE Software*, vol. 25, pp. 20–27, 2008.
2. National Vulnerability Database. http://nvd.nist.gov/.
3. Common Weakness Enumeration. http://cwe.mitre.org/.
4. OWASP. http://www.owasp.org/.
5. PCI Secutity Standard Council, Data Security Standards (PCI DSS). https://www.pcisecuritystandards.org/.
6. M. J. Kenning, “Security management standard — iso 17799/bs 7799,” *BT Technology Journal*, vol. 19, no. 3, pp. 132–136, 2001.
7. Web Application Security Consortuim Threat Classification v2.0. http://projects.webappsec.org.
8. Secure Development Standards. http://securityinnovation.com/products/teammentor/.

SecTro: A CASE Tool for Modelling Security in Requirements Engineering using Secure Tropos

Michalis Pavlidis and Shareeful Islam

School of Computing, IT and Engineering, University of East London, UK
m.pavlidis@ieee.org, shareeful@uel.ac.uk

Abstract. Secure Tropos is an extension of Tropos methodology, which considers security throughout the whole development process. The main concept of Secure Tropos is the security constraint that captures constraints regarding security. Similarly, the concepts of dependency, goal, task, resource, and capability were also extended with security in mind. In this paper we present the SecTro tool, a CASE tool that guides and supports the developers in the construction of the appropriate models of Secure Tropos.

Keywords: Security, goal modelling, requirements engineering, Secure Tropos, CASE tools.

1 Introduction

As the use of information systems is increasing rapidly everyday in finance, military, education, health care, and transportation, the need of security is increasing respectively. The stored information in many cases is sensitive and has to be secured by protecting it from any attack. In other words, there should be cost effective and operationally effective protection from undesirable events [1].

It is already agreed by the industry and research community, that security has to be considered from the early phases of the software development process [2]. Having defined the security requirements along with the functional requirements will enable the better comprehension of the system's security issues and limit the conflicts between the security and functional requirements for more secure information systems [1].

Secure Tropos is a security requirements engineering methodology that considers security throughout the whole development process [1]. The approach identifies, models and analyses the security issues from the early stages of software development within the organization and social settings [2]. But, the fact that it considers security from the early stages of software development, results in a serious increase of the activities in the software development stages and therefore requires the existence of a software tool to support the development process [2]. This paper demonstrates a tool, named SecTro, which assists the security analysts in constructing the relevant Secure Tropos diagrams that are required in order to identify, model and analyze the security issues.

The rest of the paper is structured as follows. Section 2 is a review on Secure Tropos. Section 3 illustrates the tool that supports Secure Tropos. Section 4 discusses the related work while section 5 concludes the paper and presents future work.

2 Secure Tropos Methodology

Secure Tropos is an extension of Tropos methodology that takes security into account and is based on the concept of security constraint. Also, the Tropos concepts of dependency, goal, task, resource, and capability were also extended with security in mind and formed the secure entities [1, 3]. Secure Tropos includes the following modelling activities, the security reference modelling, the security constraint modelling, the secure entities modelling, and the secure capability modelling. In addition, it consists of four stages, the early requirements, the late requirements, the architectural design, and the detailed design stages. The metamodel of Secure Tropos [4] is shown in Fig. 1 and for a more detailed description of Secure Tropos please refer to [1], [3].

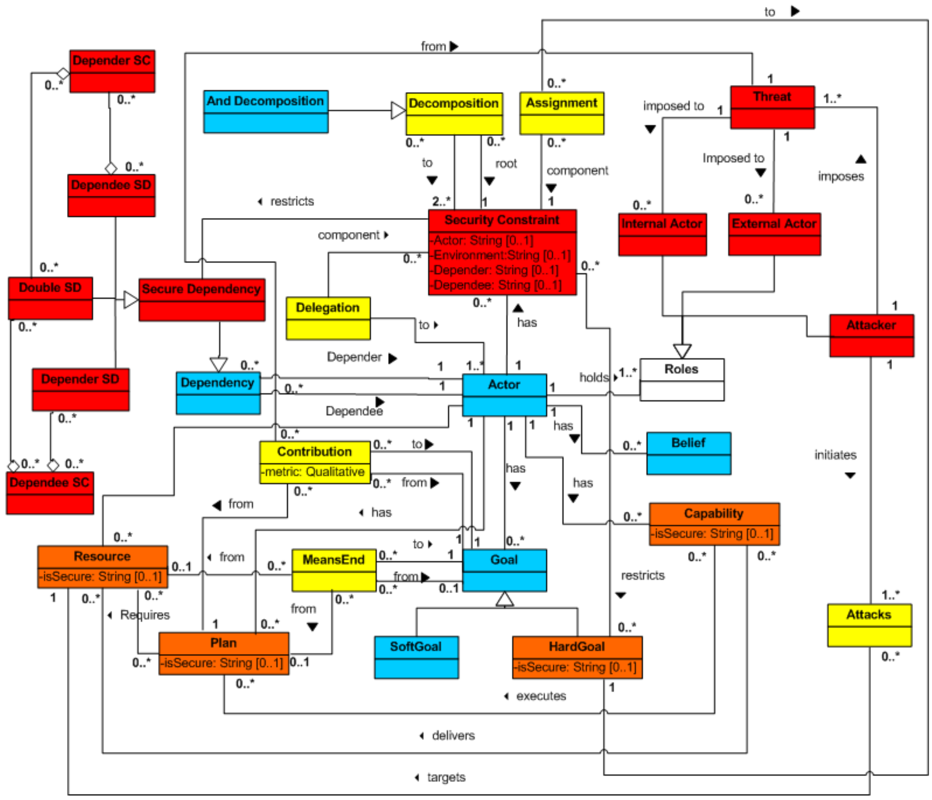


Fig.1. Secure Tropos metamodel.

3 The SecTro Tool

3.1. SecTro Architecture

SecTro is a standalone application that was built with the Java programming language making it a portable application across different platforms. The package diagram is shown in Fig. 2 and descriptions of the packages are given in Table 1. The class diagram of the classes that are responsible for the drawing functionality of the tool is shown in Fig. 3. In the ElementType class belong all the elements that can be drawn, such as an actor and a hard goal, and in the LinkType class belong all the links between the elements, such as the "plays" link and the "satisfies" link. The class diagram of the graphical user interface (GUI) package is shown in Fig. 4.

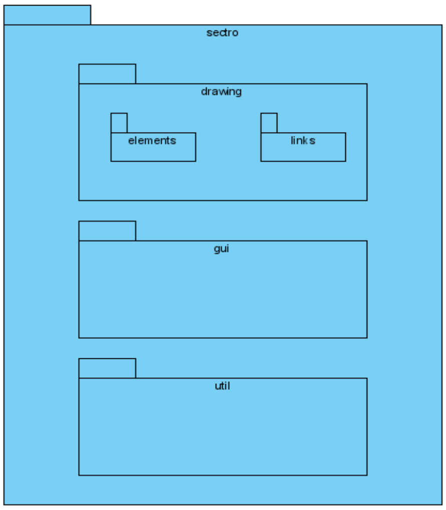


Fig. 2. Package diagram of SecTro.

Table 1. Description of the SecTro packages.

Package	Description
sectro	The parent package that includes the main class and all the sub packages
sectro.drawing	Contains the generalized class for all the drawing objects (DrawingObject) and the elements and links packages
sectro.drawing.elements	Contains the classes for all the drawing elements (Actor, HardGoal,Resource, Plan, etc.)
sectro.drawing.links	Contains the generalized class for all the Links (Link) and the classes for all the drawing links (LinkDependency, LinkRestricts, LinkPlays, etc.)
sectro.gui	Contains all the classes related to the user interface (MainForm, ToolBar, MenuBar, etc.)
sectro.util	Contains all the utility classes (ImageUtil, XMLUtil, FileUtil, etc.)

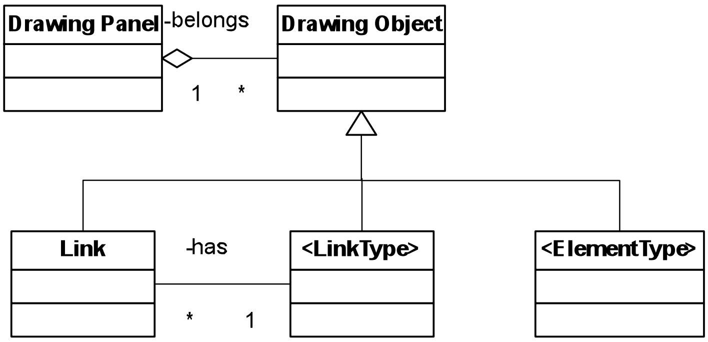


Fig. 3. Class diagram of the SecTro drawing functionality.

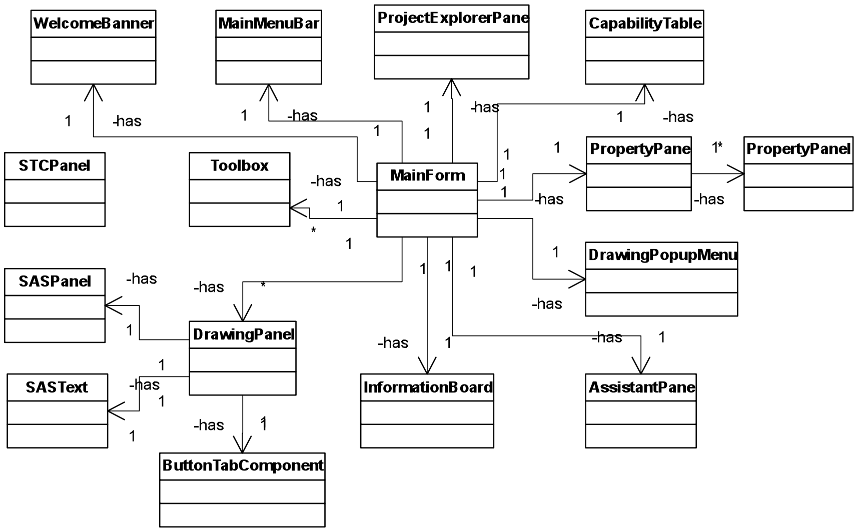
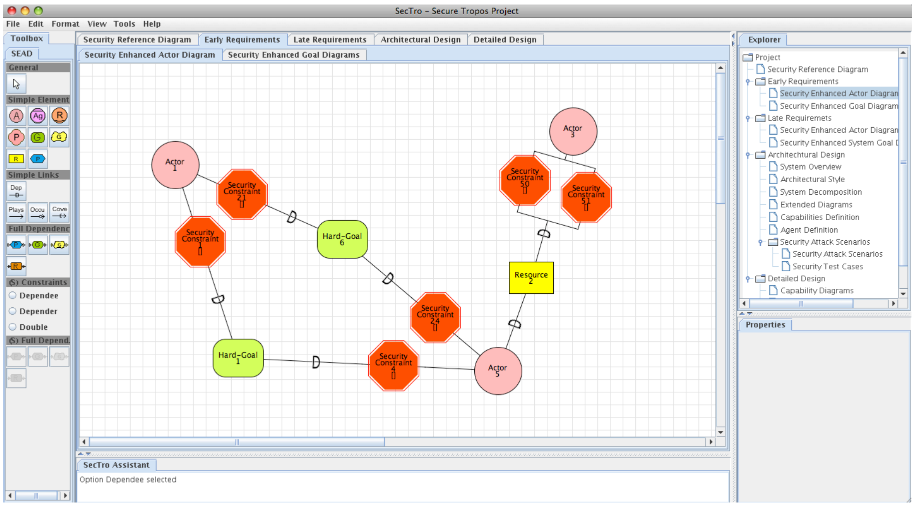


Fig. 4. SecTro GUI class diagram.

3.2. SecTro Layout and Functionalities

SecTro's workspace (Fig. 5) consists of the drawing canvas in the centre, on the top there is a series of tabs for showing the developed diagrams for each stage of Secure Tropos, the project explorer and the properties panel are on the right side, the toolbox (Fig. 6) is on the left side, and the SecTro assistant at the bottom of the workspace. The graphical representations of all the concepts of Secure Tropos by the SecTro tool are shown in Fig. 7 and the graphical representation of the secure dependency is shown in Fig. 8.



Option Dependee selected

Fig. 5. SecTro workspace.

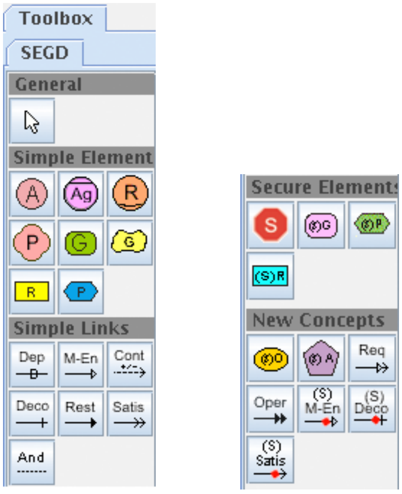


Fig. 6. SecTro toolbox.

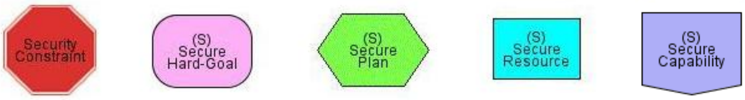


Fig. 7. Secure Tropos notation.

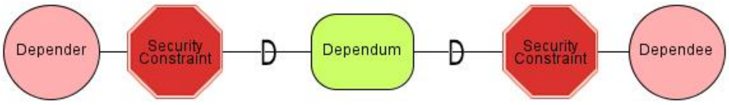


Fig. 8. Secure Dependency.

The main functionalities of the SecTro are to support the developer in the modelling activities of Secure Tropos. Therefore, the tool enables the developer to perform security reference modelling (Fig. 9), security constraint modelling (Fig. 10), secure entities modelling (Fig. 11), and secure capability modelling. During these activities the tool has a mechanism for checking the rules and constraints and informs the developer for any error. Also, the SecTro assistant panel shows more information about the rules and constrains, the concepts and the meta-models. In this way it assists the developer in the learning process of Secure Tropos methodology. Furthermore, the tool enables the developer to export the diagrams as images and in XML format.

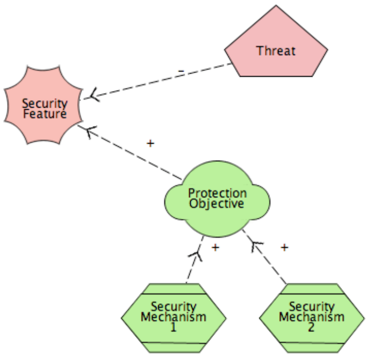


Fig. 9. Security reference modelling.

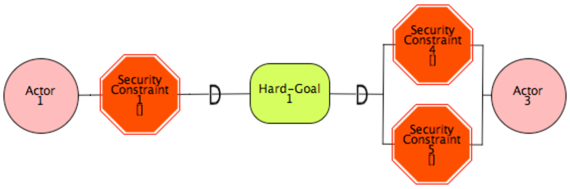


Fig. 10. Security constraint modelling.

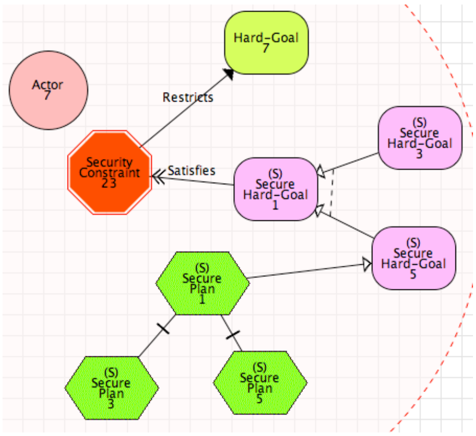


Fig. 11. Secure entities modelling.

During the architectural design the architecture of the system is defined. The tool can automatically generate the architecture style and the system decomposition. However, the activities of the architectural design can be a very difficult task for a developer without knowledge of security. Finally, in most cases, during the end of the architectural design the security attack testing takes places, where the design of the system is tested against the security requirements [5]. The tool automatically generates for the developer the security attack scenario template and the security test case template.

4 Related Work

Although Secure Tropos is still in research and it is difficult to develop a CASE tool for a methodology that is still in research, the i* modelling framework has been out for some years and a number of related CASE tools were developed to support it. OME [6], OpenOME [7], REDEPEND-REACT [8], TAOM4e [9], GR-Tool [10], T-Tool [9], ST-Tool [11], J-PRiM [12], jUCMNav [13], SNet Tool [14], and DesCARTES [15] are some examples of such tools.

The aforementioned tools, although they were developed for different ultimate purposes, they all provide support for the i* modelling framework, which is the modelling framework that was adopted by Secure Tropos as well. But, Secure Tropos introduces new concepts that none of the previous tools enables their graphical representation, i.e. security constraint, secure goal, secure plan, secure resource, and secure capability. Also, the previous tools don't provide support for the modelling activities that Secure Tropos introduces, i.e. security constraint modelling, secure entities modelling, and secure capability modelling. So, despite the fact that experienced users with Secure Tropos can make conventions and use the previous tools to construct single diagrams; these tools are not adequate to support the Secure Tropos methodology.

5 Conclusions and Future Work

The tool supports the developers in the modelling activities of the early and late requirements and architectural design stages of Secure Tropos by assisting them in the construction of the relevant concepts and models that are required during the new modelling activities. Its user-friendly interface makes it easy to use and assists security analysts who are not familiar with the methodology, by providing them with information about the methodology concepts, stages, and metamodels. Also, it enforces rules and constraints and provides valuable feedback on various actions of the developers in an interactive way. The tool has already been used by the students of university of East London to model and analyse security issues of a real industry case study. However, the tool does not support the modelling activities of the detailed design stage and we consider this as future work. In addition, future work includes the extension of the XML Schema in order to validate more models of the methodology.

References

1. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), pp. 285-309 (2007)
2. Mouratidis, H., Giorgini, P.: Integrating Security and Software Engineering: Future Vision and Challenges. In: Mouratidis, H., Giorgini, P. (eds.) Integrating Security and Software Engineering: Advances and Future Visions. Idea Group Publishing, London (2007)
3. Giorgini, P., Mouratidis, H., Zannone, N.: Modelling Security and Trust with Secure Tropos. In: Mouratidis, H., Giorgini, P. (eds.) Integrating Security and Software Engineering: Advances and Future Visions. Idea Group Publishing, London (2007)
4. Matulevicious, R.: Summary of Secure Tropos Metamodel. Internal Report, University of Namur (2008)
5. Mouratidis, H., Giorgini, P.: Security Attack Testing (SAT) – Testing the Security of Information Systems at Design Time. Journal of Information Systems 32, pp. 1166-1183 (2007)
6. OME3, http://www.cs.toronto.edu/km/ome/
7. OpenOME, https://se.cs.toronto.edu/trac/ome/
8. Grau, G., Franch, X., Maiden, N.: REDEPEND-REACT: An Architecture Analysis Tool. In: 13th IEEE International Conference on Requirements Engineering, pp. 455-456. Paris (2005)
9. Morandini, M., Nguyen, C.D., Perini, A., Siena, A., Susi, A.: Tool-supported Development with Tropos: The Conference Management System Case Study. In: Luck, M., Padgham, L. (eds.) AOSE 2007. LNCS, vol. 4951, pp.182-196, Springer, Heidelberg (2008)
10. Giorgini, P., Mylopoulos, J., Sebastiani, R.: Goal-Oriented Requirements Analysis and Reasoning in Tropos Methodology. Journal of Engineering Applications of Artificial Intelligence 18(2), pp. 159-171 (2005)
11. Massaci, F., Mylopoulos, J., Zanone, N.: Computer-Aided Support for Secure Tropos. Journal of Automated Software Engineering 14(2), 341-364 (2007)
12. Grau, G., Franch, X., Avila, S.: J-PRiM: A Java Tool for a Process Reengineering i* Methodology. In: 14th IEEE International Conference on Requirements Engineering, pp. 359-360. Minneapolis (2006)
13. Mussbacher, G., Amyot, D.: Assessing the Applicability of Use Case Maps for Business Process and Workflow Description. In: 2008 International MCETECH Conference on e-Technologies, pp. 219-222. Montreal (2008)
14. Gans, G., Lakemeyer, G., Jarke, M., Vits, T.: SNet: A Modeling and Simulation Environment for Agent Networks Based on i* and ConGolog. In: Proceedings of the 14th International Conference on Advanced Information Systems Engineering, pp. 328-323 (2002)
15. UCL/ISYS - DesCARTES Architect, http://www.isys.ucl.ac.be/descartes/index.php

Advanced Management of Research Publications based on the Lightroom Paradigm

Matthias Geel, Michael Nebeling, Stefania Leone, and Moira C. Norrie

Institute for Information Systems, ETH Zurich
CH-8092 Zurich, Switzerland
{geel|nebeling|leone|norrie}@inf.ethz.ch

Abstract. Adobe Lightroom is an example of a domain-specific information management tool that has adopted many of the practices that have become well-established within the research community such as automatic metadata extraction, fast keyword-based retrieval and query-based collections for the management of images. It also supports the workflow of professional photographers. We will present a web-based system that transfers some of these advanced concepts to the realm of research publications. The new application introduces sophisticated facilities for classifying publications in different dimensions based on combined notions of flagging, tagging, ratings and colours in addition to manual organisation and sorting in collections. Retrieval and discovery in a potentially large publication corpus is achieved by providing efficient browsing techniques based on a faceted search interface and user-driven categorisation.

Keywords: domain-specific information management tools, faceted browsing, search interface

1 Introduction

Adobe Lightroom[1] is an example of a domain-specific information management tool for digital photographs. Many professional photographers use it to browse and manage large collections of digital photographs as well as performing post-production activities in an efficient way [9]. In order to support the management of thousands of pictures, it has adopted many of the practices that have become well-established within the research community such as automatic metadata extraction, fast keyword-based retrieval and query-based collections. Another popular example is Apple's iTunes[2] which provides similar features for managing music and video files. Both systems incorporate advanced organisation and retrieval techniques considered as emerging research trends in information management and human-computer interaction such as dynamic, faceted search [1, 2, 4, 14], sophisticated tagging systems [3, 11, 12] and the integration of web resources [7].

[1] http://www.adobe.com/products/photoshoplightroom

[2] http://www./apple.com/itunes

We aim to transfer the ideas and concepts introduced by state-of-the-art information management tools for digital media to other domains and different types of personal information. As a first step, we have developed a system for the management of research publications in the form of PDF documents and their associated metadata. Our system provides a faceted search interface similar to Adobe Lightroom, but adapted for effectively browsing research publications by taking into account both available metadata specific to publications as well as user-defined criteria. We will show how the system can support researchers in tasks and workflows related to scientific publishing, for example, when carrying out a literature review in order to collect and manage related work.

We will start with an overview of related work in Sect. 2, followed by a presentation of the use case that motivates the system and supported workflows in Sect. 3. In Sect. 4, we will outline the interaction with the system and describe the key elements of the user interface. Section 5 gives an overview of the implementation and our demonstration is summarised in Sect. 6.

2 Background

As well as supporting advanced means of organising and searching for data, Adobe Lightroom [9] differs from most information management tools in terms of its extensive support for the entire *information workflow* from the capture and storage of images, through the organisation and processing of them, to the publishing of images in a variety of formats such as slideshows and web pages. In addition, images can be published to Web 2.0 sites such as Facebook and Flickr. Existing publication management systems such as Mendeley Desktop or EndNote also provide search facilities and tools for managing publications using categories, labels and favourites. However, in constrast to Lightroom, they do not provide different views for different workflows and neither adapt the user interface to the current task the user is performing, nor do they provide helpful abstractions in different phases of a larger management and review process.

Other systems designed to improve information retrieval through the use of semantic data include Haystack [8] and iMecho [2]. Haystack allows users to store references to arbitrary objects of interest along with any other properties, i.e. attributes and relationships, that they consider to be important. These properties can then be used to support both faceted querying and associative browsing. iMecho takes these ideas further by building associative links between resources from implicit access patterns of user activity sequences. In addition to supporting various kinds of search services, it is important that personal information management systems can help users *organise* their information. The basic hierarchical model of the file system and the desktop metaphor continue to dominate even though many studies have shown that users often struggle to organise their resources in a way that suits their activities [15]. While many alternatives have been proposed over the years, including the document piles of Lifestreams [5] and the collection-based approach of MyLifeBits [6] that allows an information item to be categorised into arbitrary collections, these have had little influence on desktop systems. However, alternatives to the hierarchical model of organisation can be seen, not only in various Web 2.0 applications such as Flickr, but also within desktop applications such as Adobe Lightroom.

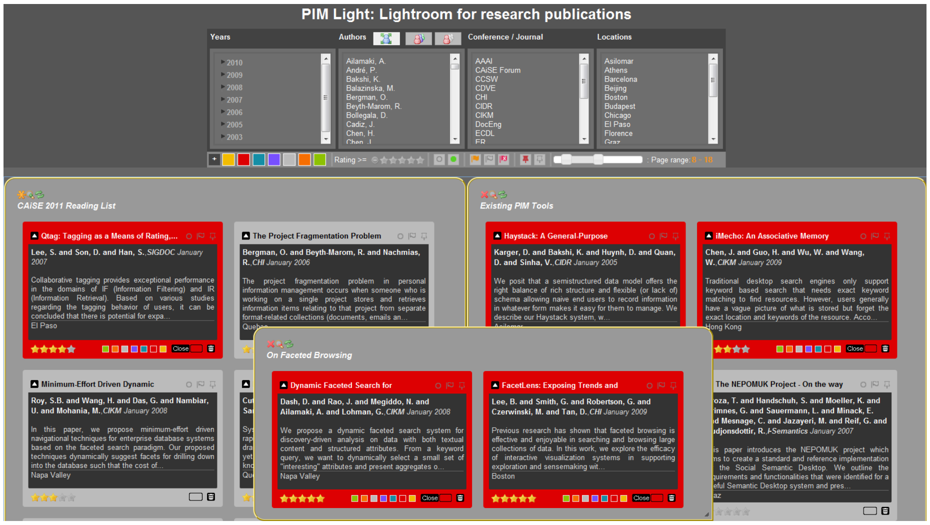
While studies proclaim tagging as a viable substitute for file-based document management [12], so far no adequate user interface solution that supports a transition away from traditional document management has been proposed. One of the predominant problems with manual tagging is the increased effort and cognitive load for users. Facets on the other hand are usually easier to deal with than free-form tags, especially when they can be extracted semi-automatically. Note that facets do not have to be text-based, but can also incorporate colours, groups or numeric values such as ratings. Faceted search is still a very active area of research and several innovative visualisations have been proposed, e.g. [10, 13].

3 System and Use Case

We will illustrate the Lightroom information management paradigm at the core of our system by considering the example of managing scientific publications for a literature review. Researchers typically gather and compile their reading lists of relevant publications from various sources. They then read and classify them according to relevance and other attributes based on personal preferences in order to select the set of most relevant papers for a specific work. Given that a literature review may involve a huge number of publications, a publication management tool that provides a flexible and lightweight approach to efficient categorisation as well as fast searching and retrieval can help achieve that task more efficiently. By adopting the Lightroom model, we have designed the system illustrated in Fig. 1 that supports the management and classification of research publications along multiple degrees of freedom.

Figure 1 shows a screenshot of the application as we have used it to gather related work for this particular paper. The publications managed by the system have been imported from existing BibTeX files so that the system has access to the publication metadata such as *title*, *author*, *booktitle*, *year* and *location*. The tool offers a faceted search bar at the top where the bibliography can be filtered according to various attributes. For example, we can quickly filter and display only those publications that have been published at CAISE in previous years, and then continue to filter by authors as well as a specific year or location.

In the centre, all publications that meet the search criteria are displayed as boxes, with the title at the top followed by an excerpt of the abstract and additional metadata. In our example, we have organised the publications into three collections. The collection on the left-hand side contains all publications that are part of the reading list compiled for this paper. The collection in the middle is a hand-picked selection of publications that are related to the topic of faceted browsing and the collection on the right contains papers about existing personal information management tools. These collections were created directly in the browser simply by dragging and dropping publication entries from one



Qtag: Tagging as a Means of Rating,...
Lee, S. and Son, D. and Han, S.,*SIGDOC January 2007*
Collaborative tagging provides exceptional performance in the domains of IF (Information Filtering) and IR (Information Retrieval). Based on various studies regarding the tagging behavior of users, it can be concluded that there is potential for expa...
El Paso

The Project Fragmentation Problem
Bergman, O. and Beyth-Marom, R. and Nachmias, R.,*CHI January 2006*
The project fragmentation problem in personal information management occurs when someone who is working on a single project stores and retrieves information items relating to that project from separate format-related collections (documents, emails an...

Haystack: A General-Purpose
Karger, D. and Bakshi, K. and Huynh, D. and Quan, D. and Sinha, V.,*CIDR January 2005*
We posit that a semistructured data model offers the right balance of rich structure and flexible (or lack of) schema allowing naive end users to record information in whatever form makes it easy for them to manage. We describe our Haystack system, w...

iMecho: An Associative Memory
Chen, J. and Guo, H. and Wu, W. and Wang, W.,*CIKM January 2009*
Traditional desktop search engines only support keyword based search that needs exact keyword matching to find resources. However, users generally have a vague picture of what is stored but forget the exact location and keywords of the resource. Acco...
Hong Kong

Minimum-Effort Driven Dynamic
Roy, S.B. and Wang, H. and Das, G. and Nambiar, U. and Mohania, M.,*CIKM January 2008*
In this paper, we propose minimum-effort driven navigational techniques for enterprise database systems based on the faceted search paradigm. Our proposed techniques dynamically suggest facets for drilling down into the database such that the cost of...
Napa Valley

Dynamic Faceted Search for
Dash, D. and Rao, J. and Megiddo, N. and Ailamaki, A. and Lohman, G.,*CIKM January 2008*
We propose a dynamic faceted search system for discovery-driven analysis on data with both textual content and structured attributes. From a keyword query, we want to dynamically select a small set of "interesting" attributes and present aggregates o...
Napa Valley

FacetLens: Exposing Trends and
Lee, B. and Smith, G. and Robertson, G. and Czerwinski, M. and Tan, D.,*CHI January 2009*
Previous research has shown that faceted browsing is effective and enjoyable in searching and browsing large collections of data. In this work, we explore the efficacy of interactive visualization systems in supporting exploration and sensemaking wit...
Boston

The NEPOMUK Project - On the way
roza, T. and Handschuh, S. and Moeller, K. and rimnes, G. and Sauermann, L. and Minack, E. d Mesnage, C. and Jazayeri, M. and Reif, G. and udjonsdottir, R.,*I-Semantics January 2007*
is paper introduces the NEPOMUK project which ns to create a standard and reference implementation the Social Semantic Desktop. We outline the quirements and functionalities that were identified for a eful Semantic Desktop system and pres...
az

Fig. 1. Publication system populated with the related work from this paper

collection to another. Note that the faceted search bar on top filters the content of all collections, which is useful when searching across multiple publication libraries.

All publications currently displayed can also be directly manipulated using the in-place editing controls for colouring and rating at the bottom of each box. In our example, we have used these features to colour important publications in red and rate them according to relevance, as can be seen by the stars at the bottom of each entry. Making combined use of multiple and orthogonal classification dimensions allowed us to classify the relevant publications and quickly choose the ones to be referenced in this paper.

4 Lightroom Paradigm for Publication Management

The most important step in designing the proposed system was to decide which concepts from Lightroom are suitable for the management of research publications and how they would translate to the new domain and different metadata. Figure 2 highlights the key features of our search interface that align it with the use of metadata and faceted search in Lightroom, but it also illustrates the changes we had to make when moving from digital photographs to research publications.

Similar to Lightroom, we distinguish between metadata and user-driven attributes. Metadata can be extracted directly from the BibTeX source and define the publications in terms of various dimensions which can then be used as facets in faceted browsing. Suitable candidates for facets are attributes where the same values appear multiple times and ideally cluster the information space into sig-

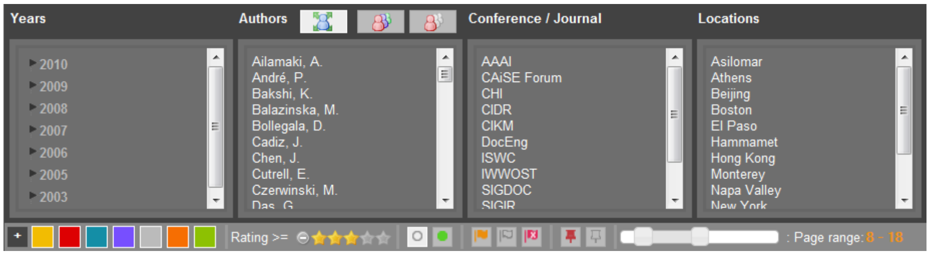


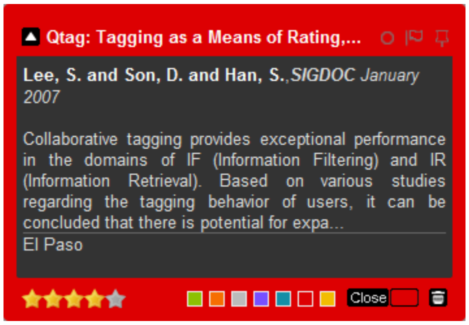
Fig. 2. Faceted search interface for research publications

nificantly large groups. For each of these facets, the user interface offers a list of attribute values inferred from the data that can be selected and combined to perform faceted browsing. In terms of research publications, the candidates we have chosen are *Authors*, *Conference/Journal*, *Locations* and *Year* (top of Fig. 2). Note that the publication title is not a suitable candidate as the title is usually unique and better used in combination with keyword-based retrieval.

On the other hand, user-driven attributes introduce a way for users to classify publications, which allows them to quickly organise, manage and browse their publication corpus. Examples of user-driven attributes in our system include colours, ratings and different kinds of flagging, e.g. to mark read/unread papers. These user-driven attributes can then also be queried with corresponding toggle buttons and sliders that hide or show matching publications (bottom of Fig. 2).

The main advantage of faceted browsing is that users can only perform selections that actually yield results. Additionally, facets can be re-calculated after every refinement to immediately give the user some feedback about the result size of further selections. All selections and restrictions are performed in real-time and the results are immediately shown to the user. When designing the faceted search interface for the publication domain, it was important to consider the types of queries a user may want to perform. Since publications are often the combined effort of more than one author, our system provides several search modes for the *Authors* facet. In the first mode, several authors can be selected simultaneously, selecting all the publications that have been written by any of the selected authors. The second mode is similar to the first one, but performs a conjunction of selected authors resulting in all publications that have been written by selected authors collaboratively. The last mode allows publications to be filtered by first author only.

Another important difference when moving to research publications relates to how Lightroom visualises the contents of photo collections where it makes extensive use of thumbnails that can be directly created from the pictures. For other types of information, it is typically required to dynamically look up or even create thumbnails derived from metadata, e.g. album art for MP3s or individual frames from movies. While, in our case, the generation of thumbnail images from the PDF is technically possible, it was considered impractical as the scaled down version of a text document might be barely readable and often provides only a snapshot of the first page. In addition, the two-column layout used by many research publications provides very few visual cues that might help users remember a particular publication based on its thumbnail. That is why we opted for an approach where we create an appropriate representation based on the publication's metadata such as title, authors and abstract. In Fig. 3, an example of our visual representation of a single publication is given. These "thumbnails" are enhanced with small, unobtrusive user interface elements which enable users to directly manipulate some key attributes as mentioned before.



Qtag: Tagging as a Means of Rating,...

Lee, S. and Son, D. and Han, S.,*SIGDOC January 2007*

Collaborative tagging provides exceptional performance in the domains of IF (Information Filtering) and IR (Information Retrieval). Based on various studies regarding the tagging behavior of users, it can be concluded that there is potential for expa...

El Paso

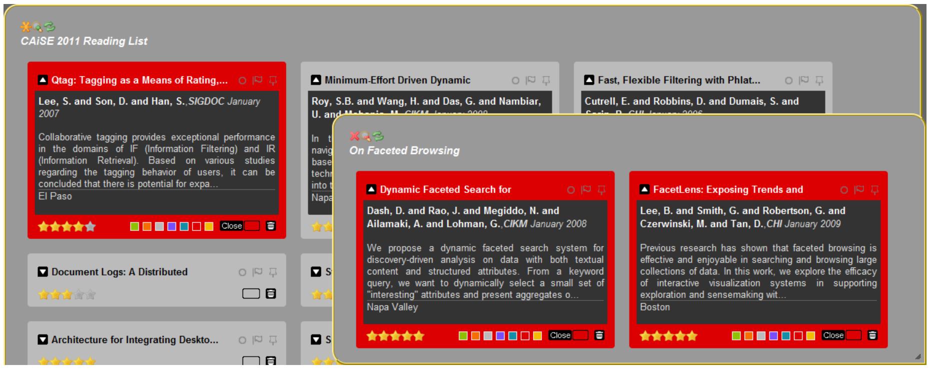
Fig. 3. Visual representation of a single publication with direct attribute manipulation

Finally, our system also supports the creation and maintenance of different collections of research publications. These can, for example, be used to maintain personal or shared reading lists, individual publication lists as well as all publications of a research group. Our solution allows ad-hoc lists to be quickly created that can be laid out spatially as illustrated in Fig. 4. Using this approach, users are free to arrange the publications according to their preferences since individual publications can be moved, not only within a collection, but also between collections using simple drag-and-drop interactions.

5 Implementation

The system is based on a rich client web architecture and was created based on the popular jQuery web framework[3] in order to support rich and responsive interactions as well as a look-and-feel similar to the original *Adobe Lightroom*. The server-side components were implemented based on CakePHP[4]—a PHP web development framework with well-suited abstractions along the MVC design pattern—and are responsible for user account registration and login as well as the storage and retrieval of publication collections. Facet calculation and filtering in the client is based on AJAX and HTML DOM manipulation techniques

[3] http://www.jquery.com
[4] http://www.cakephp.org



CAiSE 2011 Reading List

Qtag: Tagging as a Means of Rating,...

Lee, S. and Son, D. and Han, S.,*SIGDOC January 2007*

Collaborative tagging provides exceptional performance in the domains of IF (Information Filtering) and IR (Information Retrieval). Based on various studies regarding the tagging behavior of users, it can be concluded that there is potential for expa...

El Paso

Minimum-Effort Driven Dynamic

Roy, S.B. and Wang, H. and Das, G. and Nambiar,

Fast, Flexible Filtering with Phlat...

Cutrell, E. and Robbins, D. and Dumais, S. and

Document Logs: A Distributed

Architecture for Integrating Deskto...

On Faceted Browsing

Dynamic Faceted Search for

Dash, D. and Rao, J. and Megiddo, N. and Ailamaki, A. and Lohman, G.,*CIKM January 2008*

We propose a dynamic faceted search system for discovery-driven analysis on data with both textual content and structured attributes. From a keyword query, we want to dynamically select a small set of "interesting" attributes and present aggregates o...

Napa Valley

FacetLens: Exposing Trends and

Lee, B. and Smith, G. and Robertson, G. and Czerwinski, M. and Tan, D.,*CHI January 2009*

Previous research has shown that faceted browsing is effective and enjoyable in searching and browsing large collections of data. In this work, we explore the efficacy of interactive visualization systems in supporting exploration and sensemaking wit...

Boston

Fig. 4. Management and sorting of research publications in collections

that allow us to reduce loading time of publication entries and dynamically show and hide them according to whether they match the search criteria. We have also developed a number of tools for converting BiBTeX to XML and RSS formats, allowing for easy import of existing bibliographies as well as exporting publication lists managed with our system. Moreover, a lightweight version of the system, with only faceted search rather than publication management capabilities enabled, can be integrated with existing web sites, e.g. those of research groups, and allow visitors to quickly filter and browse publication databases.

6 Demonstration

In our demonstration, we will show the publication management system described in this paper. We will provide several example publication databases, including personal and shared reading lists for different topics, but we will also give the opportunity for interested parties to explore the novel management facilities using their own bibliographies imported from BibTeX. In this way, visitors will be able to experience a Lightroom-like publication management system for themselves and test whether it enables them to browse and search individual or groups of publications more efficiently.

Acknowledgements

This work was supported by the SNF under research grant 200021_121847.

References

1. Basu Roy, S., Wang, H., Das, G., Nambiar, U., Mohania, M.: Minimum-Effort Driven Dynamic Faceted Search in Structured Databases. In: Proc. of 17th ACM Conf. on Information and Knowledge Management (CIKM 2008) (2008)

2. Chen, J., Guo, H., Wu, W., Wang, W.: iMecho: An Associative Memory based Desktop Search System. In: Proc. 18th ACM Conf. on Information and Knowledge Management (CIKM'09) (2009)
3. Cutrell, E., Robbins, D., Dumais, S., Sarin, R.: Fast, Flexible Filtering with Phlat. In: Proc. of Conf. on Human Factors in Computing Systems (CHI 2006) (2006)
4. Dash, D., Rao, J., Megiddo, N., Ailamaki, A., Lohman, G.: Dynamic Faceted Search for Discovery-Driven Analysis. In: Proc. of 17th ACM Conf. on Information and Knowledge Management (CIKM 2008) (2008)
5. Freeman, E., Gelernter, D.: Lifestreams: a Storage Model for Personal Data. SIGMOD Record 25(1) (1996)
6. Gemmell, J., Bell, G., Lueder, R.: MyLifeBits: a Personal Database for Everything. Comm. ACM 49(1) (2006)
7. Groza, T., Handschuh, S., Moeller, K., Grimnes, G., Sauermann, L., Minack, E., Mesnage, C., Jazayeri, M., Reif, G., Gudjonsdottir, R.: The NEPOMUK Project - On the way to the Social Semantic Desktop. In: Proc. I-Semantics'07 (2007)
8. Karger, D., Bakshi, K., Huynh, D., Quan, D., Sinha, V.: Haystack: A General-Purpose Information Management Tool for End Users Based on Semistructured Data. In: Proc. Intl. Conf. on Innovative Data Systems Research (CIDR 2005) (2005)
9. Kim, G.K.: Early Research Strategies in Context: Adobe Photoshop Lightroom. In: Proc. ACM Intl. Conf. on Human-Computer Interaction (CHI Extended Abstracts 2007) (2007)
10. Lee, B., Smith, G., Robertson, G., Czerwinski, M., Tan, D.: FacetLens: Exposing Trends and Relationships to Support Sensemaking within Faceted Datasets. In: Proc. of Conf. on Human Factors in Computing Systems (CHI 2009) (2009)
11. Lee, S., Son, D., Han, S.: Qtag: Tagging as a Means of Rating, Opinion-Expressing, Sharing and Visualizing. In: Proc. of 25th Annual ACM International Conf. on Design of Communication (SIGDOC 2007) (2007)
12. Oleksik, G., Wilson, M., Tashman, C., Mendes Rodrigues, E., Kazai, G., Smyth, G., Milic-Frayling, N., Jones, R.: Lightweight Tagging Expands Information and Activity Management Practices. In: Proc. of Conf. on Human Factors in Computing Systems (CHI 2009) (2009)
13. Smith, G., Czerwinski, M., Meyers, B., Robbins, D., G. Robertson, G., Tan, D.: FacetMap: A Scalable Search and Browse Visualization. IEEE Trans. on Visualization and Computer Graphics 12(5) (2006)
14. Wilson, M., André, P.: Backward Highlighting: Enhancing Faceted Search. In: Proc. of 21st annual ACM Symposium on User Interface Software and Technology (UIST 2008) (2008)
15. Zacchi, A., Shipman, F.: Personal Environment Management. In: Proc. 11th European Conf. on Research and Advanced Technologies for Digital Libraries (ECDL 2007) (2007)

Diagen: A Model-driven Framework for Integrating Bioinformatic Tools

Maria José Villanueva, Francisco Valverde, Ana Levín, and Oscar Pastor

Centro de Investigación en Métodos de Producción de Software
Universitat Politècnica de València
Camino de Vera S/N 46022, Valencia, Spain
{mvillanueva, fvalverde, alevin, opastor}@pros.upv.es

Abstract. Nowadays, the diagnosis of disease based on genomic information is feasible by searching genetic variations on DNA sequences. However, geneticists struggle with bioinformatic tools that are supposed to simplify DNA sequence analysis. As a universal tool to support every requirement is far from be implemented, geneticists themselves must solve the data exchange among several tools. Due to the fact that there are no standards to support this integration task, it must be managed in every analysis. This paper proposes addressing this integration by means of a model-driven framework. The Diagen framework is a software implementation based on conceptual modeling principles that formalizes data exchange and simplifies bioinformatic tool integration. First, we analyze how conceptual modeling can be used to deal with data exchange among tools. And then, as a proof of concept, the presented framework is used to search for variations on the BRCA2 gene using real DNA samples and a set of specific bioinformatic tools.

Keywords: Model-Driven Development, Tool Integration, DNA sequence analysis

1 Introduction

Recent genetic discoveries have opened the door to personalized disease diagnosis based on DNA sequence analysis. Nowadays, it is possible to predict the risk of getting a certain disease by searching for specific genetic variations on the DNA sequence [1].

Geneticists perform DNA sequence analysis aided by bioinformatic tools. Even though these tools are functional and useful for reducing time and complexity, none of them completely fulfill all the geneticists' requirements [2]. As a consequence, geneticists are forced to use several tools in order to gather all the functionality and, eventually, accomplish the complete DNA sequence analysis.

One important issue regarding these tools is that data exchange among them is required. The problem lies in the fact that each of these tools is isolated and uses its own data format to report the computed information. For this reason, data exchange among tools is a non-trivial task that geneticists must address in each analysis according to the following procedure: 1) Export data from the source tool; 2) Understand the semantics of the tool-specific data format; 3) Perform a translation into the target tool format; and finally, 4) Import the data into the target tool.

As geneticists usually lack Software Engineering knowledge, most of them perform this task manually or develop programming scripts. Although these specific scripts are useful in solving minor problems, they are far from being compliant with good practices of Software Engineering. The implemented scripts to support data exchange are often coupled solutions that integrate only two specific tools. In the end, these solutions cannot be reused and compromise the geneticists flexibility for using other tools.

As a solution, this paper proposes the application of conceptual modeling to develop a model-driven framework that formalizes data exchange and simplifies tool integration. In order to provide a high quality solution, this work has been developed in the context of a collaboration with geneticists from the Genomic Medicine Institute (IMEGEN). As a proof of concept, the proposed framework integrates several tools that are used by IMEGEN geneticists in their daily routine to search for genetic variations using real DNA samples of the BRCA2 gene (a gene related to Breast Cancer).

The paper is organized as follows: Section 2 presents a brief summary of other proposed solutions to solve the tool integration problems in DNA sequence analysis. Section 3 explains the proposed model-driven framework for integrating bioinformatic tools. Section 4 presents how the framework is used for disease diagnosis support using samples of the gene BRCA2 and a set of bioinformatic tools. And finally, section 5 presents the conclusions and future work.

2 Related Work

Several works have attempt to overcome current DNA sequence analysis tool issues. These proposals follow two different approaches.

Several sequence file formats for expressing bioinformatic tools results have emerged. Examples of these formats are: 1) Variant calling formats, such as the Variant Call Format (VCF) proposed for the 1000 Genomes Project [3]; 2) Alignment results formats, such as the Sequence Alignment/Map Format (SAM) [4], which provides a compressed textual representation, and the Genome Variation Format (GVF) [5], which provides a textual format using the Sequence Ontology [6].

All these formats have been defined for the purpose of providing interoperability among different DNA sequence analysis tools. The implementation of decoupled data exchange mechanisms is feasible using any of the above examples as a standard format. However, their main drawbacks are the complexity of each textual format and the mandatory implementation of a low- level mechanism to extract the data. As a consequence, none of them have become a widely applied standard and are only used in the research context where they have been proposed.

Several bioinformatic development frameworks have also been implemented. Some examples of these frameworks are Biojava [7], BioPython [8], or BioPerl [9]. These frameworks provide an API that supports common functionality for DNA analysis tasks. Additionally, they provide several format conversion operations to transform file formats among different tools.

These frameworks have been defined to provide geneticists with the freedom to implement their personalized tools. However, the geneticists still have to worry about low-level programming details and integration issues.

3 An Integrative Framework for Bioinformatics

This work presents a model-driven framework for the integration of DNA sequence analysis tools and retrieval of genetic information. Diagen is classified as a model-driven framework because each of its components (classes, data entities, operations) is a projection of the Conceptual Schema of the Human Genome (CSHG) [10]. The CSGH is a conceptual model created with the collaboration of geneticists, where biological concepts related to the human genome have been precisely addressed and defined. The framework uses this conceptual model to support the following DNA sequence analysis tasks (Figure1):

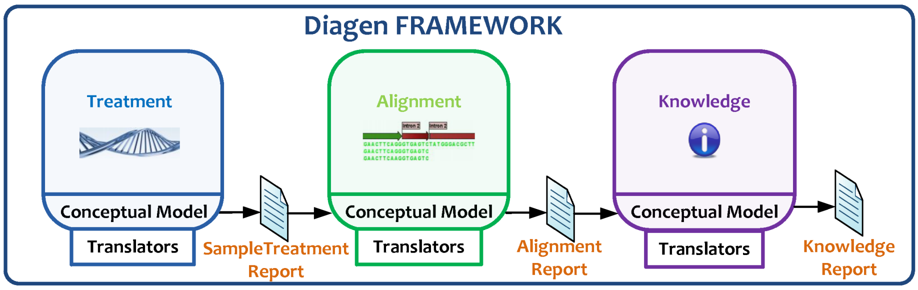


Fig. 1. General View of the Framework

1. Sequence Treatment: A DNA sequence is rebuilt from the fragments generated by the sequencing machines.
2. Sequence Alignment: A DNA sequence is aligned to a reference sequence in order to determine the differences between them.
3. Variation Knowledge: Using data gathered in genomic databases, each sequence difference that is related to a disease is reported.

Data exchange among tools is a difficult task because there is a great variety of formats to express the different results. Taking into account that data exchange is required when a tool calculates data that another tool requires, it can be assumed that both tools must share a set of common concepts. Therefore, it is possible to define a conceptual model that represents those shared concepts and establishes well-defined boundaries and vocabularies.

Diagen establishes the common context to guide data exchange among tools that define a conceptual model for each task transition:

- The Sample Treatment Report conceptual model (Figure 2) defines all the concepts related to the reconstructed sequence in the sequence treatment task (T1) to be analyzed in the sequence alignment task (T2).

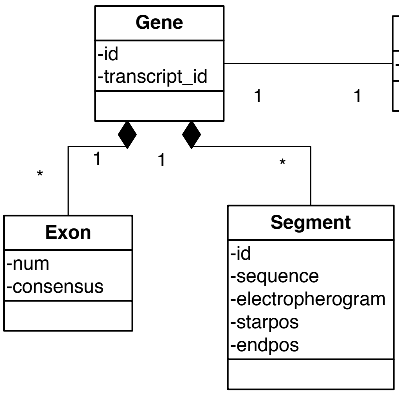


Fig. 2. Sample Treatment Report Conceptual Model

- The Alignment Report conceptual model (introduced in [11]) defines all the concepts related to the differences found in the sequence alignment task (T2) to be characterized in the variation knowledge task (T3).
- The Knowledge Report conceptual model (Figure 3) defines all the concepts related to the characterized variations to be used for other task (for example, a diagnosis report creation task).

Data exchange among tools that perform these tasks usually requires the implementation of a translation mechanism to understand each other. In that case, data expressed in a concrete format needs to be translated into a different format. However, the use Diagen avoids these coupled implementations because a tool to be integrated in the framework only needs a translator that expresses its outputs in terms of the underlying conceptual model. This translator is easier to implement since it only requires establishing the relationships between the output and the conceptual model.

Each task that is supported by the framework has been implemented to be independent from the others, and, therefore, it can be used separately. Thanks to this modularity, it is possible, for example, to use the alignment task in another environment. In this case, the input data should be provided in terms of the input conceptual model (Sample Treatment Report) and the output report should be read in terms of the output conceptual model (Alignment Report).

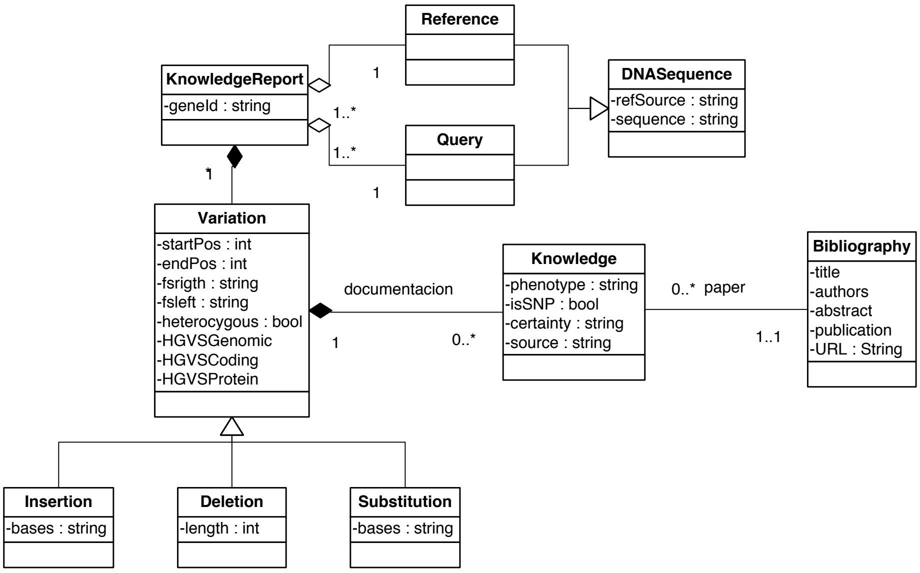


Fig. 3. Variation Knowledge Conceptual Model

The Diagen framework has been implemented using the Java language. Additionally, each conceptual model involved in data exchange has software correspondence with a set of Java classes and a XML representation. In order to manage both representations (Java and XML) JAXB (Java Architecture for XML bindings) [12] has been used. This is a specific API that allows Java objects to be parsed in a XML data and vice-versa.

4 Using Diagen for Disease Diagnosis Support of the BRCA2 Gene

As a proof of concept, the framework has been used to develop a prototype for disease diagnosis support of Breast Cancer. This specific framework configuration integrates several bioinformatic tools that are used daily by the geneticists of IMEGEN.

Recently, the framework (Figure 4) has been applied to integrate:

1. Sequence treatment task: The Sequencher tool [13] is used to rebuild the samples provided by a sequencing machine.
2. Sequence Alignment task: The implementation of the algorithm BLAST from NCBI [14] is used to search for differences in the sequence. There is also an integrated tool that is based on the Smith-Waterman Algorithm (SW Tool) and a tool that looks for known-variations in the sequence by aligning flanking sequences (Flanking Tool).

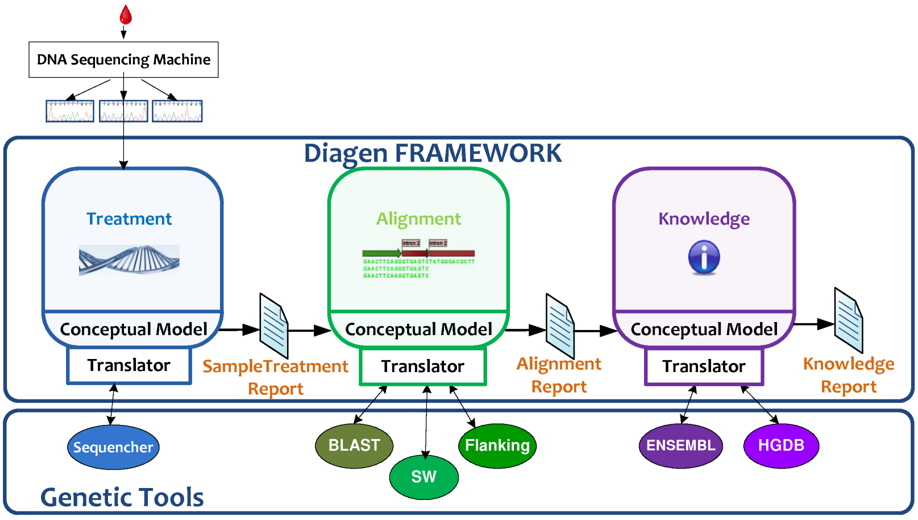


Fig. 4. IMEGEN configuration of the framework

3. Variation Knowledge task: Variation characterization is performed manually by geneticists searching in several databases. However, this framework provides two mechanisms for genetic knowledge data retrieval. The first mechanism obtains some data from the ENSEMBL database [15]. The second mechanism retrieves genetic information from the HGBD database [16] based on the Conceptual Schema of the Human Genome (CSHG) [10].

The prototype supports the three defined tasks needed to perform a DNA sequence analysis. As a result, it retrieves a personalized report containing the genetic variations and the potential diseases of the individual.

The main advantages of the framework are: 1) A decrease in the execution time, 2) A reduction in the efforts needed for data exchange among tools; and 3) The elimination of the need to search for variation data in the huge set of databases spread around the Web.

The prototype has been tested with real samples of the gene BRCA2 (Table 1). The test was carried out analyzing the BRCA2 gene sample from ten different patients (P1-P10). For each patient, the table shows the number of variations characterized by IMEGEN, the number of variations characterized by Diagen, and the accuracy that Diagen offers compared with the IMEGEN manual process. IMEGEN performs the analysis in approximately four hours (depending on the success achieved while searching for a difference in the genetic repositories).

The preliminary test showed that Diagen offers the results almost instantly and with an accuracy rate of between 60-90%. It is also important to emphasize that the variations that were not characterized by Diagen were always the same variations (7 variations in total) that appeared repeatedly in all the analyses.

Table 1. Preliminary BRCA2 tests

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Characterized Var. IMEGEN	7	10	8	8	8	13	9	10	9	8
Characterized Var. Diagen	6	6	7	6	5	8	6	7	6	5
Accuracy rate %	86	60	88	75	63	62	67	70	67	63

5 Conclusions and Future Work

This work proposes a model-driven framework that is based on a well-defined conceptual model of the human genome in order to address DNA sequence analysis. As a proof of concept, the Diagen framework is configured for the development of a disease diagnosis support and is tested by means of real DNA samples of the BRCA2 gene.

We have realized that the tools available actually accomplish some of the geneticists' goals. The problem lies in the fact that geneticists' activities, specifically in the DNA analysis domain, lack standard methodologies, well-defined tasks, fixed vocabularies, and unified knowledge sources. As a consequence, the execution of a DNA sequence analysis cannot be performed efficiently or without geneticists' intervention.

The solution to these problems is not to reinvent new DNA sequence analysis tools but to integrate the most suitable tools according to geneticists' needs. The presented framework applies conceptual modeling to integrate different bioinformatic tools and to provide a common context to exchange data with each other. The main advantage of the presented framework, over other integration approaches is that Diagen is a high-level abstraction framework that provides concise and significant tasks to geneticists instead of low-level tasks. Moreover, with this framework, geneticists can perform a DNA sequence analysis and forget about the data formats of different tools.

As genetics is a very innovative field that is constantly evolving with new discoveries, all concepts must be well-defined without ambiguity. Thanks to the conceptualization of the DNA sequence analysis tasks, all the involved concepts are precisely formalized. As a consequence, it is easier to adapt the tasks to changes or to support new concepts.

The preliminary results are promising, but there is room for improvement. The low accuracy detected is because the missed variations were not described in the integrated sources. As these sources are constantly improving, it is expected that future versions will solve these issues.

As future work, the framework will be extended to support other bioinformatic tasks. The main goal of this extension is to design a complete framework that supports other genetic functionality besides DNA sequence variation analysis. Additionally, the next step is to apply the service-oriented paradigm to provide a more flexible development environment. With this approach, geneticists could select only the required functionality, defined as services, and easily create a personalized tool.

Acknowledgments Thanks to the Instituto de Medicina Genómica (IMEGEN, http://www.imegen.es) for its support and the provision of data. This work has been developed with the support of MICINN under the FPU grant AP2010-1985 and the project PROS-Req (TIN2010-19130-C02-02), and co-financed with ERDF.

References

1. Margaret A. Hamburg and Francis S. Collins. The Path to Personalized Medicine. *New England Journal of Medicine*, vol. 363(4), pp. 301–304, (2010)
2. Nicole Rusk. Focus on Next-Generation Sequencing Data Analysis. *Nature Methods*, vol. 6(11s), pp. S1, (2009)
3. Siva Nayanah. 1000 Genomes Project. *Nat Biotech*, vol. 26(3), pp. 256–256, (2008)
4. Heng Li et al. The Sequence Alignment/Map Format and SAMtools. *Bioinformatics*, vol. 25(16), pp. 2078–2079, (2009)
5. Martin G. Reese et al. A Standard Variation File Format for Human Genome Sequences. *Genome biology*, vol. 11(8), pp. R88+, (2010)
6. Karen Eilbeck, Suzanna Lewis, Christopher Mungall, Mark Yandell, Lincoln Stein, Richard Durbin, and Michael Ashburner. The Sequence Ontology: A Tool for the Unification of Genome Annotations. *Genome Biology*, vol. 6(5), pp. R44, (2005)
7. R. C. G. Holland et al. BioJava: An Open-Source Framework for Bioinformatics. *Bioinformatics*, vol. 24(18), pp. 2096–2097, (2008)
8. Peter J. A. Cock et al. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics*, vol. 25(11), pp. 1422–1423, (2009)
9. Jason E. Stajich et al. The Bioperl Toolkit: Perl Modules for the Life Sciences. *Genome Research*, 12(10), pp. 1611–1618, 2002.
10. Oscar Pastor, Ana Levin, Matilde Celma, Juan Casamayor, Aremy Virrueta, and Luis Eraso. Model-Based Engineering Applied to the Interpretation of the Human Genome. In Roland Kaschek and Lois Delcambre, (eds.) *The Evolution of Conceptual Modeling*, LNCS, vol. 6520, pp. 306–330. Springer, Heidelberg (2011)
11. Maria Jose. Villanueva, Francisco. Valverde, and Oscar Pastor. Applying Conceptual Modeling to Alignment Tools: One Step towards the Automation of DNA Sequence Analysis. *BIOINFORMATICS 2011*, (2011)
12. E. Ort and B. Mehta. Java Architecture for XML Binding (JAXB). Technical Report Sun Developer Network, (2003)
13. P Curtis C Bromberg, H Cash and CJ Goebel. *Sequencher, Gene Codes Corporation*. Ann Arbor, Michigan, 1995.
14. NCBI BLAST (Basic Local Alignment Search Tool). Availble in http://blast.ncbi.nlm.nih.gov.
15. Hubbard, T. et al. The ENSEMBL Genome Database Project. *Nucleic Acids Research*, vol. 30(1), pp. 38–41, (2002)
16. Oscar Pastor et al. Enforcing Conceptual Modeling to Improve the Understanding of Human Genome. *Research challenges in information Science (RCIS)*, pp. 85-92, (2010)

Visualizing Variability Models Using Hyperbolic Trees

R. Bashroush, A. Al-Nemrat, M. Bachrouch, H. Jahankhani

School of Computing, IT and Engineering,
University of East London,
London, United Kingdom
{rabih, ameer, hamid2}@uel.ac.uk

Abstract. Software Product Line Engineering (SPLE) has emerged in recent years as a viable way to maximize reuse when designing a family of related products. One of the main tasks conducted during the SPLE process is Variability Management (VM). VM is about identifying commonality among the different products being developed while capturing and cataloging variability. In real-life projects, VM models tend to encompass a very large number of variants reaching in many projects the order of thousands. Visualizing these models has been a major challenge for tool developers. In this work, we present our MUSA CASE tool which uses hyperbolic trees for representing VM models and supports gesture based interaction (using multi-touch interfaces). The tool has been successfully used to develop a large scale case study.

Keywords: Software Product Lines, Variability Management, Feature Modeling, Hyperbolic Trees.

1 Introduction

Software Product-line Engineering (SPLE) has emerged as a major strategy for maximizing reuse when a family of related software systems is developed. In this approach, commonality-variability analysis [1] (Variability Management - VM) of the member products is a major phase of the process and plays an important role in its success.

One of the main challenges within VM is the handling and visualizing “industry-size” models which usually comprise a large number of variability points along with the dependency relationships that exist among them. The challenge comes from the large amount of information captured within a model (business related, dependency and relationships, etc.) as well as the current techniques and I/O devices used to visualize the model which do not inherently scale [13].

The MUSA CASE tool was designed to overcome these challenges [13], [14]. MUSA is based on our successful work on multiple-perspective based variability management which provides a rich modeling framework while using the concept of separation-of-concerns to alleviate the problem of information overloading. MUSA implements this theory using a mind-mapping modeling approach, hyperbolic trees, over the state-of-the-art in HCI, the multi-touch Microsoft Surface [2]. This provides a scalable solution that taps on the latest in Natural User Interface (NUI) [3] design providing an intuitive and large display for VM. In addition, the MUSA solution provides interfaces over other multi-touch platforms including Windows 7 (using its native multi-touch support).

The theory behind MUSA is highlighted in section 2. An overview of the MUSA CASE tool is then presented in section 3. Finally, section 4 ends with related work and conclusion.

2 Technical Background

The Four Views Model (4VM) forms the theoretical foundation upon which MUSA is designed as a Proof-of-Concept. The original version of the 4VM can be found here [4] and to appear here [5].

It is generally agreed that different stakeholders have interest in considering different views of the product line variability model [4],[6]. So, it is important for a VM mechanism to be able to extract and present relevant information about the family model in dedicated views for different groups of stakeholders (users, system analysts, developers, etc.). This could considerably contribute to alleviating the graphical overload when showing all the information in one view (as compared to using multiple views). This is one of the core concepts behind 4VM.

The 4VM proposes a four view presentation of the feature model which are discussed below.

2.1 Business View

The Business View is aimed at the project business and management stakeholders. It acts as a portal for inputting and presenting information related to:

- *Feature implementation time*: This indicates when a given feature is to be implemented. Planning for future releases of products, the features to be implemented in these products, and the timing, is a key step for the success and sustainability of a product line
- *Feature Cost/Benefit analysis*: information related to the effort needed and cost involved in realizing features as well as their foreseen benefit. This provides valuable input to the overall project costing and the product versioning process
- Open/Closed sets of features: it is rarely the case that the architect is furnished with a system's comprehensive and complete set of features upfront. Rather, features are continuously added (and modified) to the initial feature model over time. Designing a system around an open and changing set of features is a very challenging task. To overcome this problem, some industries designate some features as closed, meaning that they can't be changed (core features), while others are designated as open, meaning they can be modified by developers.
- Negative features: these are the features that are not mean to be supported by the system (e.g. for security reasons) as opposed to supported features.

These properties are usually specified and used by the project managers to carry out system-wide business analyses which support decision making such as when to introduce features within a product line; what features are feasible from a business perspective, etc.

2.2 Hierarchical & Behavioral View

The Hierarchical and Behavioral View is the view provided by most existing feature modeling techniques. In this view, information related to the structure of the feature model and the behavior of the features is captured. Among other potential users, this view is mainly targeted at architects and developers.

2.3 Dependency and Interaction View

Due to the size and complexity of feature dependency and interaction within real-life systems, a separate view is created within the 4VM to model these relationships. The Dependency and Interaction View is complementary to the Hierarchical and Behavioral View. We define feature dependency and feature interaction as follows:

- *Feature Dependency*: a feature-to-feature dependency where the inclusion of one or more features affects one or more features within the system.
- *Feature Interaction*: a feature-to-architecture dependency where the inclusion of one or more features affects the architecture structure (different component sets and/or configurations, etc.).

In this view, logic design is proposed to capture the dependency and interaction relationships. Once the relationships are modeled, standard logic algorithms (and SAT solvers) can be used to simplify the models.

2.4 Intermediate View

Finally, the intermediate view has been introduced in an attempt to bridge the gap between feature modeling and the architecture design. This gap exists between the two domains due to the fact that the feature model is based on end-user and stakeholder concerns while the architecture structure is designed to accommodate technical concerns.

To bridge this gap, the intermediate view proposed attempts at injecting design decisions into the feature model to take it one step further towards the architecture domain. As such, it may be regarded as an intermediate stage between feature model and system architecture.

3 Implementation

MUSA was funded as a Proof-of-Concept project to demonstrate the theoretical foundation provided in 4VM. The MUSA system provides an end-to-end variability management solution as shown in Figure 1 below. MUSA provides a rich and collaborative interface to elicit and manage requirements and variability from stakeholders while allowing for appropriate access to the variability model to different teams including: implementation, testing and deployment teams. In addition, MUSA automates model verification (with the use of SAT solvers) and maintains consistency among the different views with the help of a centralized Database (as shown in Figure 1). MUSA is considered among the very first CASE tools to move into the NUI space in order to overcome scalability issues.

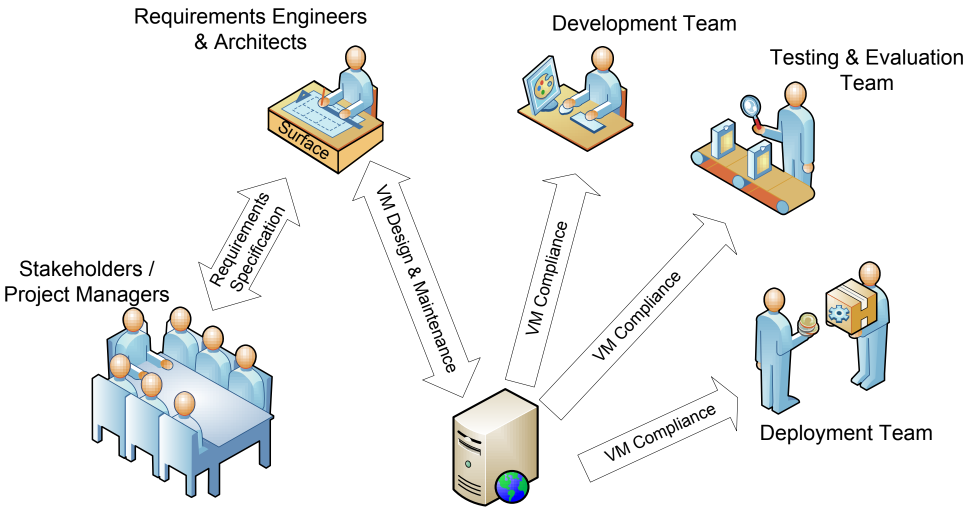


Fig. 1. The end-to-end MUSA System overview

For example, with MUSA, users can user different gestures such as: pinching (for expanding nodes), panning (by moving two fingers on the screen to shift the model), three finger gesture (to center the model at the root node), etc.

In addition, one of the main advantages of MUSA over other CASE tools within the domain of VM (see next section) is scalability. This is made possible with the adoption of hyperbolic trees [15] to represent VM rather than normal trees and other structures within the Euclidean space.

Hyperbolic trees (a.k.a. hypertree) is a visualization method that maps graphs into the hyperbolic geometry. The result effect is similar to a fish-eye lens view where nodes in focus are placed in the center and given more room, while out-of-focus nodes are compressed near the boundaries. Focusing on a different node brings it and its children to the center of the screen, while compressing out of focus nodes.

The advantage of this is that the standard tree suffers from visual clutter when the number of child nodes grow exponentially (in the order of 2^n for binary trees and much quicker for other types of trees), thus, requiring an exponential amount of space to be displayed appropriately. However, hyperbolic trees employ hyperbolic space which provides more room compared with Euclidean space. This is because increasing objects' size in Euclidean space would cause objects to increase linearly in size compared to hyperbolically in the hyperbolic space [15].

Figure 2 (using the MS Surface) and figure 3 (using Windows 7) below show an example VM of a case study developed with the MUSA toolset. In these figures, we notice color coding is used to distinguish between optional (blue) and mandatory features (yellow).

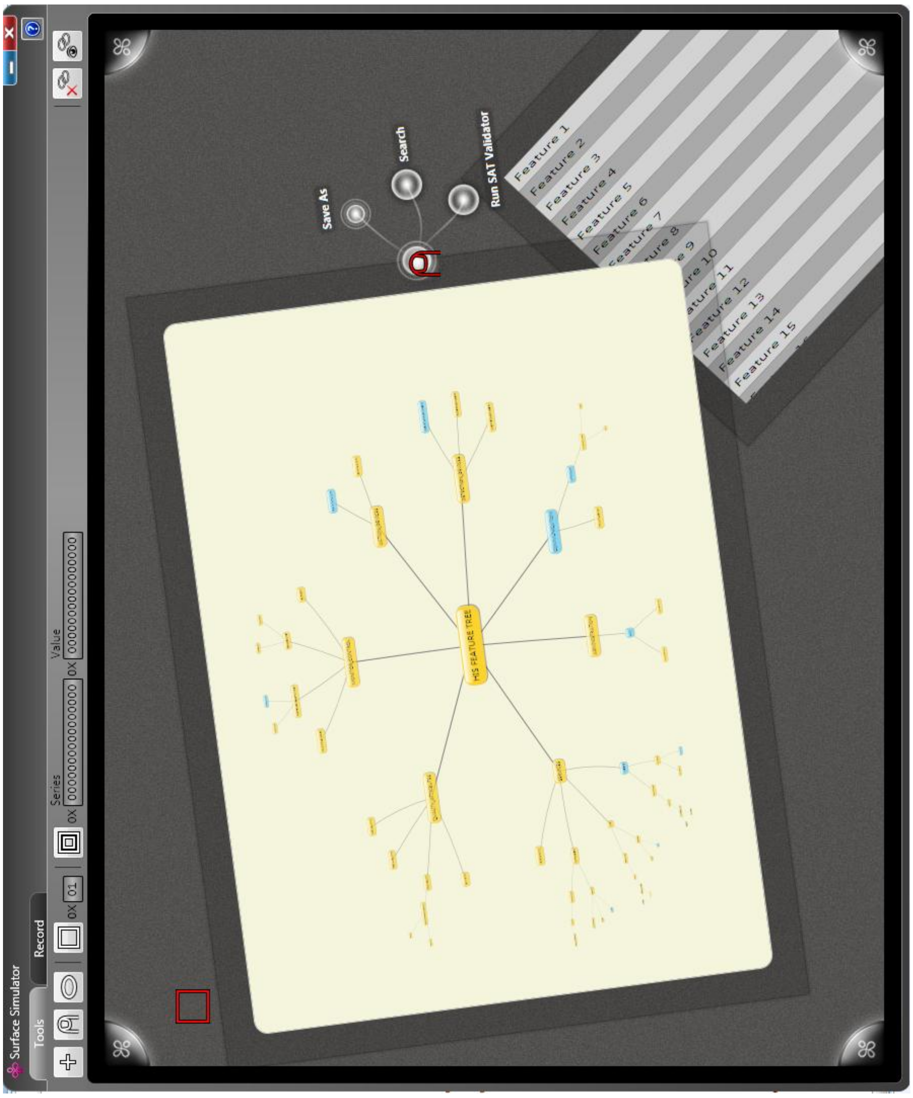
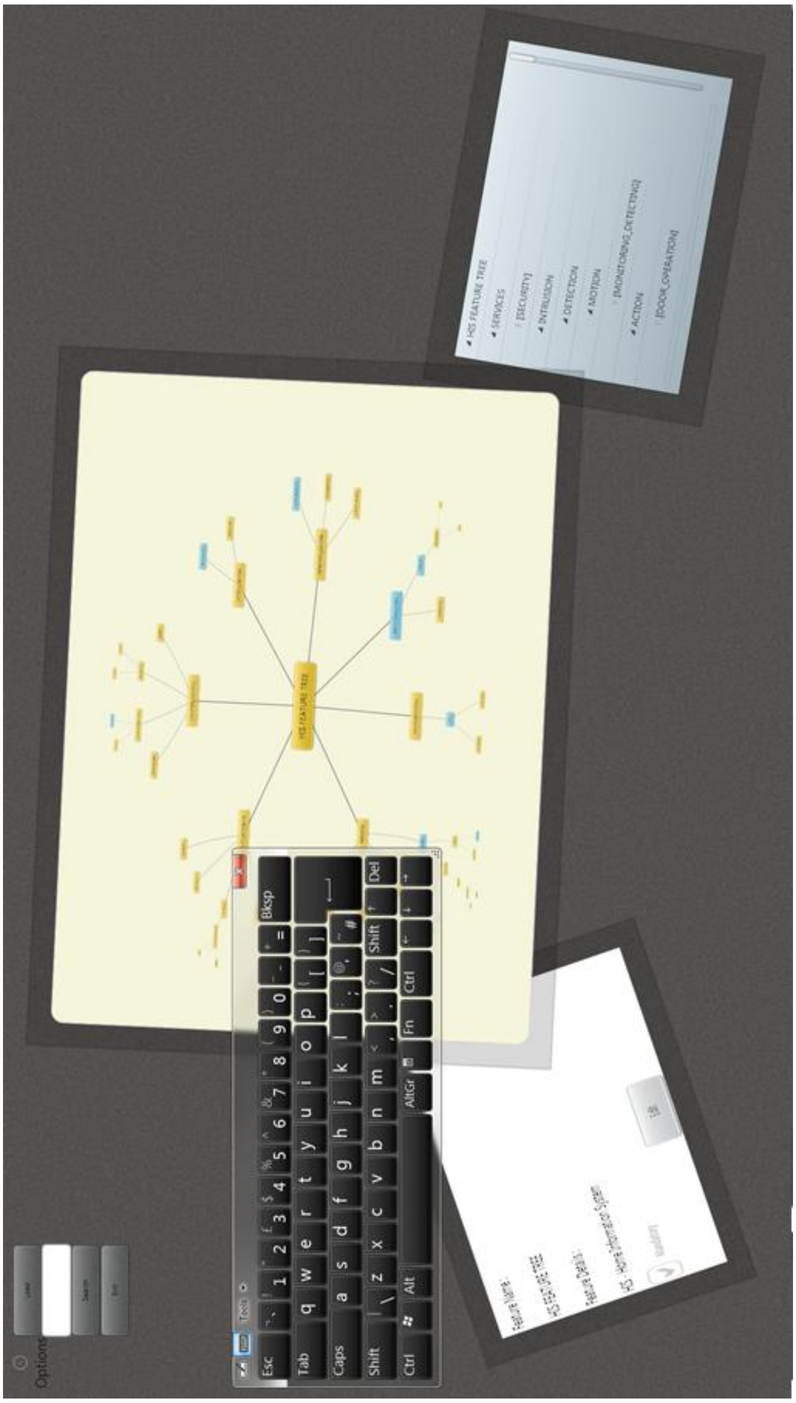


Fig. 2. MUSA over the MS Surface Interface



Feature Name:
HIS FEATURE TREE
Feature Details:
HIS - Home Information System
Mandatory

Fig. 3. MUSA over the Windows 7 Interface

4 Conclusion and Related Work

Over the past few years, a number of VM approaches have been developed ranging from research techniques to commercial products.

On the research techniques front, Sinnema et al [7] introduced the COVAMOF framework and toolset which uses the COVAMOF variability view (CVV) to represent the view of variability for the product family artefacts. The graphical notation used is based on a simple 2D, unidirectional tree that becomes cumbersome to use as soon as the number of variants exceeds about the 50. The Feature Modelling Tool [16] was created as a plugin to visual studio (Figure 4 below). Yet again, in practice, the tool would be difficult to use and manage as soon as the number of variants exceeds 60 or 70. Other tools include FeaturePlugin (an eclipse plugin) by Antkiewicz and Czarnecki [8] and Kubmang by Asikainen et al [9].

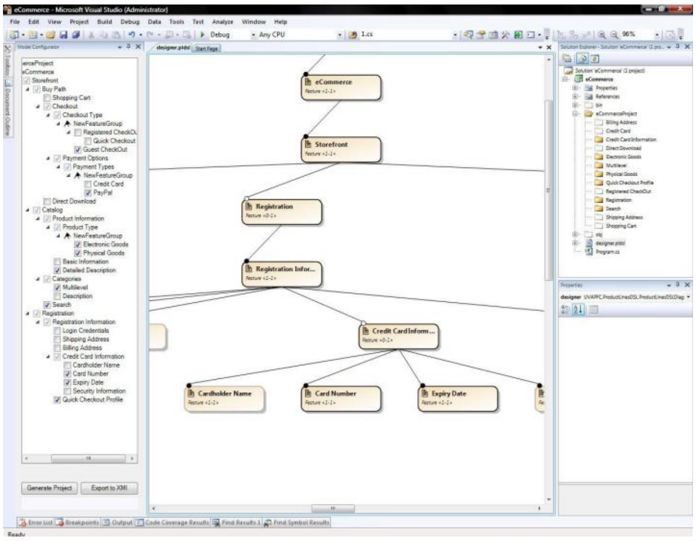


Fig. 4. Feature Modeling Tool [16]

The major challenge for most research techniques is scalability. The scalability issue arises from the graphical modeling techniques traditionally adopted (e.g. trees) and the I/O devices used (standard keyboard, mouse, and monitors). More recently, virtual reality technologies have been reported as being explored as a potential approach for VM. It is hard to see how such techniques could make their way to commercial environments due to the difficulty involved in integrating such approaches within existing industrial development settings.

On the commercial products front, the main tools are from Pure-Systems [11] who have introduced the pure::variants [10] solution and BigLever [12] who developed the Gears toolset. Both are provided as part of a complete modeling framework. These commercial products have managed scalability by largely moving away from graphical representation of models. File system tree like structures and even text listings (e.g. using MS Excel sheets) have been seen in use. Although such approaches scale and are in industrial use, adopting NUI interfaces such as the one we implemented in MUSA will increase productivity, time-to-market and allow for the creation and management of larger and more complex product families.

Acknowledgments. The work on the MUSA project has been funded by the European RD Fund through INI under the Proof of Concept funding scheme [2008-2010]. It has also received further funding under the Challenge Fund scheme at the University of East London [2010-2011]. We thank all the postgraduate students at the CITE school at UEL who contributed to some of the testing and development of the MUSA toolset as part of their thesis work.

References

1. K. C. Kang, J. Lee, and P. Donohoe, "Feature-Oriented Product Line Engineering," IEEE Software, vol. 19, pp. 58-65, (2002)
2. Microsoft Surface, http://www.microsoft.com/surface/
3. Natural User Interfaces, http://en.wikipedia.org/wiki/Natural_user_interface
4. R. Bashroush, I. Spence, P. Kilpatrick, TJ Brown, and C. Gillan. "A Multiple Views Model for Variability Management in Software Product Lines," Proceedings of the Second International Workshop on Variability Modelling of Software-intensive Systems. Essen, Germany, (2008)
5. US Patent Application No 12/349,797, Inventor: Rabih Bashroush, Title: “Multiple Perspective Feature-based Variability Management”, (Patent Pending)
6. B. Nuseibeh, J. Kramer, and A. Finkelstein, "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification," IEEE Transactions on Software Engineering, vol. 20(10), pp. 760-773, (1994)
7. M. Sinnema, S. Deelstra, J. Nijhuis, and J. Bosch, "COVAMOF: A Framework for Modeling Variability in Software Product Families." In proceedings of Third Software Product Line Conference 2004, Boston, (2004)
8. M. Antkiewicz and K. Czarnecki, "FeaturePlugin: feature modeling plug-in for Eclipse." In proceedings of the 2004 OOPSLA workshop on eclipse technology eXchange, (2004)
9. T. Asikainen, T. Männistö, and T. Soininen, "Kumbang: A domain ontology for modelling variability in software product families," Advanced Engineering Informatics, Elsevier Science Publishers B. V., vol. 21, pp. 23-40, (2007)
10. D. Beuche, "Variant Management with pure::variants," pure-systems GmbH (2003)
11. Pure-Systems Pure::Variants, http://www.pure-systems.com/Variant_Management.49.0.html
12. "BigLever Software Gears," http://www.biglever.com/solution/product.html
13. R. Bashroush. "A NUI Based Multiple Perspective Variability Modelling CASE Tool," Muhammad Ali Babar, Ian Gorton (Eds.): ECSA 2010. Lecture Notes in Computer Science, Volume (6285), Springer-Verlag Berlin Heidelberg, ISBN 978-3-642-15113-2, August 2010
14. R. Bashroush. "A Scalable Multiple Perspective Variability Management CASE Tool". Proceedings of the 14th International Software Product Line Conference (SPLC), South Korea. September 2010.
15. Lamping, John; Rao, Ramana; Pirolli, Peter (1995). "A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies". Proc. ACM Conf. Human Factors in Computing Systems, CHI. ACM. pp. 401–408.
16. Grupo de Investigacion en Reutilizacion y Orientacion a Objeto (GIRO) – Feature Modeling Tool. Available from: http://www.giro.infor.uva.es/FeatureTool.html[last visited May 2011]

AnnL: The Tool for Planning for Viable Enterprises

Edward Lewis[1]

[1] University of New South Wales, Australian Defence Force Academy, Canberra, Australia

e.lewis@adfa.edu.au

Abstract. This paper presents the Analysis of Networked Links (AnnL) tool that supports the principles and practices for the planning for viable enterprises, through such disciplines as Enterprise Engineering (including Enterprise Architecture, Governance, and Service Management). It shows how the software can be used throughout the lifecycle of Enterprise Engineering, providing synchronized reports not readily available in other tools.

Keywords. Enterprise Engineering, Systems Planning, Analysis of networked Links, Work systems

1 Introduction

This paper describes the use of a tool that can be used to plan an enterprise so that it is viable. This tool – Analysis of networked Links (AnnL) – can be used from strategic through to tactical planning; from enterprise-wide portfolio management to detailed service design. The design of the tool is based upon the experience of using it for over 15 years in preparing more than 50 strategic plans, Business Cases, tender evaluations, architecture risk analyses, and policy developments.

Every enterprise consists of a set of processes that convert inputs into outputs as products or services. These processes require people using resources to do their job within socio-technical systems, which we will call 'work systems' [1]. If the enterprise is to be viable then these work systems must be planned (decisions made and resources allocated) so that the enterprise as a whole can act in time to avoid the effect of risks or to take up opportunities.

We will consider how the AnnL tool supports general managers (as they are designers too [2]) or specialists such as strategic planners, Business Process Managers/ Decision managers, Enterprise Architects, or Service Managers - gathered under the label of Enterprise Engineering in either the European [2] or US [3] meaning- in their planning of work systems that make up viable enterprises.

There are five risks in the existing techniques and tools for Enterprise Engineering:

- Incomplete consideration of all resources – Until recently the most important resources, Who and Whom, were overlooked in the commonly used Enterprise Architecture frameworks. Recently, we have seen the emergence of Human Views, thankfully [5]. There is almost no mention of Worth in any framework, including

Zachman (as recently pointed out by [6]). Only one tool, Abacus [7], helps in determining the cost of ownership of architectures – despite the mention of Economic views in GERAM [8] (or ISO 15704, if you prefer). Even Archimate, which does provide a notation schema that applies to most of the layers (resources) in a work system, misses skills, worth, facilities, and time. We need techniques and tools that cover all of the resources.

- Insufficient consideration of alignment – Currently, Enterprise Architecting looks the wrong way. It concentrates upon integrating resources across processes - along the rows in Table 1. We do have Business Motivation Models or Enterprise Visions to give the 'big picture'. We do have diagrams that show the relationship between capabilities and functions (business services) and activities and, ultimately, data or physical resources. Unfortunately, most viewpoints cover only two layers in an architecture. It is hard to align the resources over all of the levels. There are some exceptions: service maps used in Service Management do show the top-to-bottom alignment of resources to objectives, with [9] giving such an example using Archimate notation. We need to be able to align all resources in this way.
- Lack of unified approach – There is confusion in the use of different notation, terms, and approaches. There are attempts to develop standards in notation - UML, SysML or Archimate [4] - but we still need notations that address all of the layers of planning; from business motivation to implementation plans, such as physical data models or network blueprints.
- Documentation rather than design – Although some tools, such as Systems Architect or Abacus, provide simulation, risk management, or costing support for the design of architectures, most tools are merely documentation aides (as noted also by [10]). They do little to help in planning, providing only views of the intended design for buyers and blueprints for builders.
- Addressing the wrong audience – Most Enterprise Engineering tools are intended for use by specialist enterprise modelers rather than by general managers. They produce detailed visual representations of possible systems to be considered by verbally skilled senior managers as part of their decisions about the acquisition of resources. We need to support the decision-making process of these managers rather than just feeding them incomprehensible models.

AnnL is designed to remedy all of these risks.

2 Use of AnnL Tool

All that is necessary to use AnnL is to list items (resources) then link them. AnnL uses the categories of the items, the initial estimates of their parameters, and the nature of the links between the items to prepare graphical, numerical, or verbal reports to decision makers so they can judge best what should be done.

These items and the nature of their links come from the data model in Figure 1.

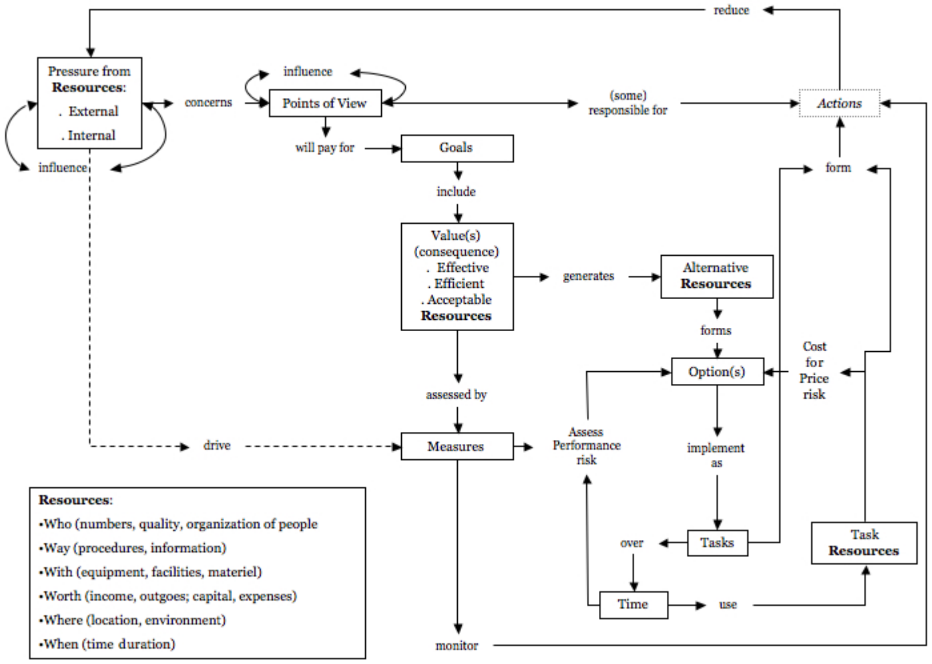


Fig. 1. Data Model for the AnnL Tool

The data model is derived from the need for AnnL to support the planning practices and the Principles of Planning that are given on the Systems Planning Mentor website at www.layrib.com, from where the software and its manual also can be downloaded.

Through this data model, AnnL enables a planner to consider the consequence of pressures that are of concern to the various key points-of-view; who are then willing to pay to have resources with requisite value (capability, capacity, constraints) to use to avoid the negative consequences and enhance the positive; to measure the risks of the options that have been generated from combinations of alternative resources thought to have the requisite capability; to determine the price risk of the options through a cost model of the resources needed to carry out the tasks that implement the options; and to describe the Action Plan for carrying out the tasks using these resources, according to blueprints (‘viewpoints’) that guide those people who are implementing that option with the least price and performance risk.

It is this list-link operation that is the key difference between AnnL and other Enterprise Architecture or data modeling tools. AnnL involves building a database (a linked list, of course) from which the diagrams or tables are generated rather than starting with a diagram and then building an ‘encyclopedia’. This approach was also advocated by [11], independently from, and well after, the inception of AnnL. Their approach does not cover the range of resources or reports as AnnL.

Abacus from Avolution [7] also uses this approach, although mostly when building an architectural description from existing lists of resources rather than during the initial design process. As an aside, AnnL could be developed into a library within

Abacus, using it as the engine for producing the Reports rather than the current use of Excel macros, but there are sufficient differences in approach, outlined below, to warrant AnnL standing alone.

List Items. So, the first action in AnnL is to list the items to be linked. Figure 2 shows an extract from the Input sheet. The planners describe each resource or action or stakeholder in the left most column. They classify each item, according to the type of resource or action, in the middle columns. Then they give initial estimates (using low-likely-high values, if necessary) in the right most columns.

Item Description	Item Type	Re-source	Sub-resource	Low	Likely	High
Grow business	Values: capable	Way	Business service	1000	1500	2500
Increase cash flow for BUP	Values: capable	Worth	Income - Earnings	500	700	
Win new business	Values: capable	Way	Business service			

Fig. 2. Input for AnnL, showing description of items, their classification, and initial estimates (extract from Excel)

In order to ensure semantic consistency, the planner classifies each item by using a pick list that draws upon the checklist of resources in the data model. The pick list in the successive columns alters according to what has been picked for the higher classification. For example, if 'Way' has been chosen as the relevant resource for a value then the sub-resources in the next column are only those part of the 'Way' checklist.

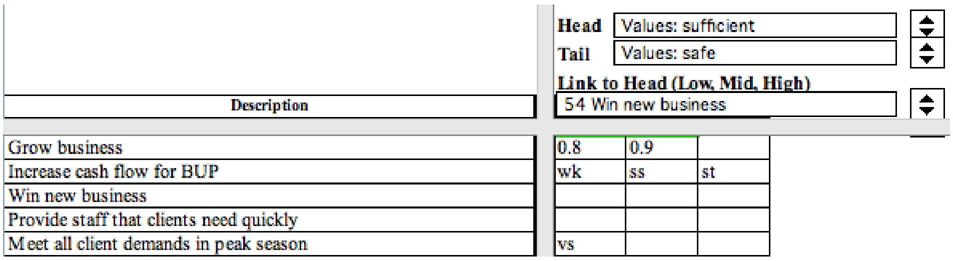
Of course, this step is easier said than done. There are many planning practices, described on the Systems Planning Mentor website, that need to be used by planners to make sure this list is correct.

Link Items. The other main action in building AnnL is to link the items. These links show the nature and extent of the relationship between the different items.

The links are made through pick lists, as shown in Figure 2. The resource at the head of the link is picked; then the resource forming the tail is picked from a list tailored to the head. AnnL then automatically moves to the rows for the tail items. Finally, the link is entered in the rows corresponding to the tails of each link, using more pick lists to ensure that the nature of the link is semantically consistent for the items being linked.

As for listing the items, the planners need to draw upon their expertise and the planning practices to determine what these links should be.

One of the differences between AnnL and other Enterprise Engineering tools is that AnnL can use, encourages the use of, verbal input. The extent of the links can be numeric (0.1 – 1) or verbal (very weak, vw, – very strong, vs), for example. Similarly, as shown later, AnnL can report numerically or verbally; whatever the recipients of the Reports prefer.



Head Values: sufficient

Tail Values: safe

Link to Head (Low, Mid, High)

54 Win new business

Description			
Grow business	0.8	0.9	
Increase cash flow for BUP	wk	ss	st
Win new business			
Provide staff that clients need quickly			
Meet all client demands in peak season	vs		

Fig. 3. List of Links, showing the pick lists that are used to link head items and row items (to the left), with the extent of the link inserted into the three columns to the right

That is all that has to be done by the planners. AnnL does the rest: carrying out the variety of analyses and producing the models that the planners need, in whatever format that they wish.

3 Applications for Enterprise Engineering

AnnL produces many different Reports, according to the application of planning. The Reports contain various versions of the trees, tables ('matrices' or 'catalogs' in TOGAF terms), or words that are the result of the calculations carried out by AnnL, using the type of items and the links between them.

One of the major advantages of AnnL is that it enables all of the many documents that make up a Business Case or the design of an Enterprise Architecture to be synchronized easily. Each time a planner changes an item or its link, AnnL will automatically update all of the Reports containing that item and link. That is, the new reports can reflect not only changes in editing of items, such as names, but also changes in the logic underlying the relationships between the components. This ability is rare in most Enterprise Architecture tools.

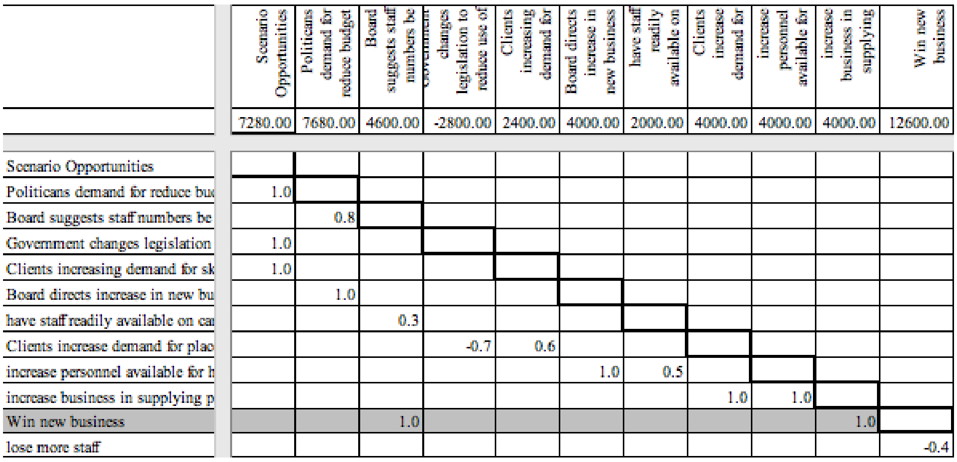
Examples of these Reports are given below. These examples show the variety of formats and results that can be produced by AnnL. The underlying algorithms used in the analyses are available from the Mentor site.

The following examples are taken from a case study (loosely) based upon an actual project. Anonymity is protected - as is the author. The case study involves a high-level strategic decision by Business Utility Providers (BUP) to take an initiative forming a new business line in hiring out specialist but redundant staff. There are a number of decisions to be made at the strategic, operational, and tactical level. After these decisions (whether to centralize or decentralize the business line or to outsource the IT, for example) have been made then AnnL can prepare the blueprints for the project managers or the systems builders.

There is not enough space to show the full portfolio of reports. The following viewpoints [12] were chosen because they show the results of techniques that only AnnL provides or bring out the versatility of AnnL or they are more compact than the diagrams that might be preferable in the actual case and so can fit in the page limit.

Consequence Chain. One of the techniques used in strategic planning is the modeling of the consequences of external or internal pressures upon the enterprise. The intention of this viewpoint is to assist planners in finding points of intervention that break the chain leading to risks or amplify the chain leading to opportunities. These interventions are values (descriptions of the required resource, such as a procedure for winning more business or a facility in a less risky location). The set of values that intervene in the chains describe the capability of the strategic initiative.

Figure 4 shows a consequence chain in the N2 or Design Structure Matrix [13] viewpoint. Of course, this example is but a very small subset of the full analysis. This viewpoint can be easier to use than the equivalent diagrammatic map for the more complicated circumstances in the usual architectural description [14]. This example shows how the value of 'Win new business' breaks the link to the risk of 'lose more staff'; nothing gets past this intervention. The numbers at the head of each column are a result of the modeling of the consequences from the cost of the risk back to the initial drivers of the risk (considering other drivers not shown in this example). So 'win new business' is an important capability.



	Scenario Opportunities	Politicans demand for reduce budget	Board suggests staff numbers be	Government changes legislation to reduce use of	Clients increasing demand for	Board directs increase in new business	have staff readily available on	Clients increase demand for	increase personnel available for	increase business in supplying	Win new business
	7280.00	7680.00	4600.00	-2800.00	2400.00	4000.00	2000.00	4000.00	4000.00	4000.00	12600.00
Scenario Opportunities											
Politicans demand for reduce bu	1.0										
Board suggests staff numbers be		0.8									
Government changes legislation	1.0										
Clients increasing demand for sk	1.0										
Board directs increase in new bu		1.0									
have staff readily available on ca			0.3								
Clients increase demand for plac				-0.7	0.6						
increase personnel available for h						1.0	0.5				
increase business in supplying p								1.0	1.0		
Win new business			1.0							1.0	
lose more staff											-0.4

Fig. 4. Consequence chain in N2 viewpoint. Intervening value shown in grey.

Influence of Points of View. Figure 5 shows the influence between the points-of-view (stakeholders). This analysis starts with estimates of the initial power of the stakeholders according to their position, entered on the list of items. AnnL combines this estimate with the extent of the influences, shown in the links, to determine who inherits the most power.

The viewpoint uses strings to represent the influence from one point-of-view to the next. The ditto (") marks represent a repetition of the item on a previous line.

The diagram shows that the PS Association representative influences the Politicians who influence the Government, and so on. There is a branch at Government, who influences both the Department of Labour and the BUP Board. The Suppliers start their own string, which intersects with the other at the BUP People CEO.

PS Prof Association [10]	Politicians[8]	Government[6]	Dept of Labour[2]	BUP Board [1]	BUP People CEO [1]	Finance Group [0]
"	"	"	"	"	"	BUP HR Department [0]
"	"	"		BUP Board [1]	BUP People CEO [1]	Finance Group [0]
Suppliers of services[5]					BUP People CEO [1]	Finance Group [0]
"					"	BUP HR Department

Fig. 5. Stakeholder Influence Diagram, using the Strings viewpoint

The numbers in brackets are the results of the calculations of cascading influence, which can be used to determine which point-of-views are key (Politicians and Suppliers in this case). AnnL does produce other tables with this information in more detail. It is dangerous politically to show this report to the actual points-of-view.

This analysis is essential for the proper design of a system [2, 5, 15] but it is not available in most Enterprise Engineering tools. Nor is the Strings viewpoint.

Values Statement. AnnL can generate values trees (or means-ends tree or Functional Decomposition Diagrams), as can most tools. More usefully, AnnL also generates verbal descriptions of values, for the senior decision-makers who are more comfortable with words than with diagrams, as shown in Table 1, AnnL uses the links between values and sub-values, and the extent to which the key stakeholders are willing to pay for each value, to determine the words ('must', 'should', ..) expressing the criticality of each value. It generates this table and the equivalent visual tree.

Table 1. Ideal State Description, listing the objectives and constraints that form the requirements of the system

In order to contribute to meeting the goal of winning new BUP business, the BUP CEO

a. Must increase awareness of staff capabilities
b. Should improve exposure to market
c. Might increase expertise of BUP staff

whilst meeting the constraints:

a. Must comply with corporate policy about …
etc

Design Analysis Display. Although Enterprise Engineering should be about design, most Enterprise Engineering tools are useful for documenting a design rather than for designing. They rarely help in creating ideas for options. AnnL extends the powerful creativity technique of (General) Morphological Analysis [16] through the use of Design Analysis Displays (see www.layrib.com).

Figure 6 shows an example of a Design Analysis Display. The diamonds are design decisions formed by each of the capability values that intervene in the Consequence Chain. The boxes are alternative solutions for these design decisions. A path through all of the alternatives is an option. In the example, there are 13 possible options.

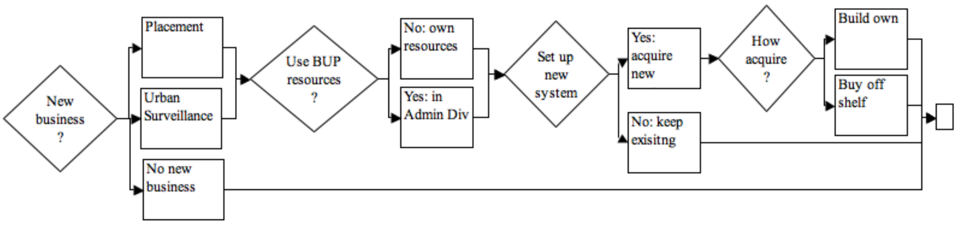


Fig. 6. Design Analysis Display

AnnL can use the judgment of the planners (shown in the links representing the assessment of the alternatives against the pertinent values) to show the 'best' paths.

Business case report. The design of work systems involves finding the option with the least performance and price risk. As shown in Table 2, AnnL produces a "Risk Picture" for decision-makers to use to trade off the performance risks of options against their total cost of ownership, adjusted for the time of expenditure. It shows the output in words or numbers, to fit to the cognitive style of the audience.

Table 2. Business Case Report, showing cost-benefit analysis, and description of risks

		Do nothing	Urban Surveillance Business	Placement Business
Price				
Income		$-	$-	$-
Outgoes		$(199,000)	$(248,000)	$(74,000)
Nett Present Value		$(199,000)	$(248,000)	$(74,000)
(Discount rate)		0%	5%	7%
Risk Cost		$2,443,182	$3,125,000	$2,386,364
Risk Adjusted Price		$(2,642,182)	$(3,373,000)	$(2,460,364)
Performance	Impact			
Grow business	essential	should not	should not	might not
Make use of staff	essential		might	
Comply with policies ..	essential	might not	should not	
Improve exposure	nice to have		might not	might not

The planner can use different financial metrics. They include cost-benefit ratios, Return on Investment, Internal Rate of Return, Nett Present Value, and (preferably) Risk-Adjusted Price – where the costs of failing to meet the values are added to the price. Abacus seems to be the only other tool to carry out such calculations.

Action Plan. AnnL can produce detailed Action Plans, as shown in Table 4. They list the tasks for implementing processes, who is responsible for the tasks, the assets needed and the budget for them, and the measurement of the quality of these tasks. This Report is in a format more familiar to managers.

AnnL also produces associated Reports. They include the list of roles, grouped over tasks; total budget for each task; and consolidated list of qualities (performance measures). These other Reports are generated at the same time as the Action Plan.

Table 3. Action **Plan (incomplete)**

	Who does	**receives**	**Task**	**Detail**	**Assets**	**Budget**	**Time**	**Quality**
1.1	BUP Board	BUP People CEO	Approve Placement Business				Year 1	
1.2	BUP People CEO HR Group	PS Professional Association Project staff	Notify staff of change in business	Pay off opponents			0.5 Year 1 0.5 Year 2	90% staff informed in 24 hours
1.3	Finance Group	Suppliers of services	Buy equipment on eBay	Get guidelines	Opera tors	-15000	Year 2	

There is feedback between the Action Plan and the Business Case. The Action Plan shows the task and resources needed to implement an option. The costs of these resources automatically appear in the pricing of the options in the Business Case.

Implementation Reports. AnnL can produce a number of reports that are useful builders of systems. Some Reports are to evaluate design options. Other Reports are produced after the Business Case has been accepted, as it is not very useful to worry about modeling something so we can do it better, if we do not need to do it at all.

These Reports could include a variety of Enterprise Architecture viewpoints. It is intended to extend AnnL to be able to produce all of the TOGAF viewpoints and the new AusDAF viewpoints (which include all of the MODAF v1.2.004 and DODAF 2.02 viewpoints plus some of its own).

4 Conclusion

The AnnL tool supports all of the planning needed to ensure an enterprise is viable. It uses a comprehensive checklist (WHAT) to provide a complete and consistent consideration of the resources at the various steps in the planning process.

All the planners need to do is to list and link items. AnnL then produces the Reports that support the design of architecture, in the formats that are most acceptable to the various audiences. These Reports are readily updated and synchronized.

If Enterprise Engineers use AnnL when planning work systems then they are free to use their insight and experience, systematically and with full analytical support, without being caught up in the drudgery of documentation.

References

1. Alter, S.: The Work System Method: Connecting People, Processes, and IT for Business Results. Work Systems Press, Larkspur, CA (2006)
2. Boland R. and Collopy, F. (eds): Managing as Designing, Stanford University Press, Stanford CA (2004)
3. Hoogervorst, P: Enterprise Governance and Enterprise Engineering. Springer (2009)
4. Rebovich, G.: Engineering the Enterprise. 1st Annual IEEE Systems Conference, Honolulu, HA, 9-13 April, p1-6 (2007)
5. The Open Group: ArchiMate 1.0 Specification. http://www.archimate.org (2009), viewed 25 Jun 2010
6. Systems Engineering and Assessment: The Human View Handbook for MODAF. Bristol, UK, Crown (2008)
7. Simsion, G.: What is wrong with the Zachman Framework?. The Data Administration Newsletter, www.tdan.com/view-articles/5279/, (2005), viewed 17 Mar 2011
8. Avolution: Abacus Products page. http://www.avolution.com.au/products.html, (2011), viewed 20 May 2011
9. Bernus, P.: Generalised Enterprise Reference Architecture and Methodology. http://www.cit.griffith.edu.au/~bernus/taskforce/geram/versions/geram1-6-3/v1.6.3.html, (1999), viewed 26 Jun 2010
10. Holschke, O., Narman, P., Flores, W., Eriksson, Evaline, and Schonherr: Using Enterprise Architecture Models and Bayesian Belief Networks for Failure Impact Analysis. J. Enterprise Architecture, 6(2), 7-18 (2010)
11. Martinez, C., Cane, Sheila, Salwa, A., Smith, K., and Lee, Kristin: *Application of Network Visualization to Identify Gaps in Complex Information System Architectures*. Tracking Number 08-0109, MITRE Corporation, VA (2008)
12. ISO/IEC 42010 FCD: Systems and software engineering – Architecture Descriptions, International Organization for Standardization. Geneva, Switzerland (due 2011)
13. Design Structure Matrix home page: http://dsmweb.org (2011), viewed 20 May 2011
14. Ghoniem, M., Fekete, J., and Castagliola, P.: A Comparison of the Readability of Graphs using Node-Link and Matrix_Based Representations. IEEE Symposium on Information Visualization, Oct 10-12, Austin, TX (2004)
15. Clegg, C.: Sociotechnical principles for systems design. App Ergon, 31, 463-477 (2001)
16. Ritchey Consulting: Swedish Morphological Society home page. http://swemorph.com (2011), viewed 20 May 2011

Stepwise Context Boundary Exploration Using Guide Words

Naoyasu Ubayashi[1] and Yasutaka Kamei[1]

Kyushu University, Japan
ubayashi@acm.org, kamei@ait.kyushu-u.ac.jp

Abstract. Most requirements elicitation methods do not explicitly provide a systematic way for deciding the boundary of the usage context that should be taken into account because it is essentially difficult to decide which context element should be included as the system requirements. If a developer explores the context boundary in an ad-hoc manner, the developer will be faced with the *frame problem* because there are unlimited context elements in the real world where the target system exists. There are many application domains that should take into account the *frame problem*: security, safety, network threats, and user interactions. To deal with this problem, this paper proposes a new type of requirements analysis method for exploring the context boundary using guide words, a set of hint words for finding a context element affecting the system behavior. The target of our method is embedded systems that can be abstracted as a sensor-and-actuator machine exchanging the physical value between a system and its context. In our method, only the *value-context elements*, a kind of *value objects*, are extracted as the associated context elements. By applying the *guide words*, we can explore only a sequence of context elements affecting the data value and avoid falling into the *frame problem* at the requirements analysis phase.

Keywords: Context analysis, Frame problem, Embedded systems.

1 Introduction

Many embedded systems not only affect their context through actuators but also are affected by their context through sensors. The term *context* refers to the real world such as the usage environment that affects the system behavior.

In most cases, context is only roughly analyzed in comparison to functional or non-functional system requirements. As a result, unexpected behavior may emerge in a system if a developer does not recognize any possible conflicting combinations between the system and its context. It is also difficult to decide the boundary of the context that should be taken into account: which context element, an object existing outside of the system, should be included as the targets of requirements analysis. If a developer explores the context boundary in an ad-hoc manner, he or she will be faced with the *frame problem* [7] because there are unlimited context elements in the real world where the system exists. The *frame problem* is the problem of representing the effects of the system behavior in logic without explicitly specifying a large number of conditions not affected by the behavior.

To deal with the *frame problem* in embedded systems, we propose CAMEmb (Context Analysis Method for Embedded systems), a context-dependent requirements analysis method. A context model is constructed from the initial system requirements by using the *UML Profile for Context Analysis*. This context model clarifies the relation between a system and its context. In CAMEmb, only the *value-context elements*, a kind of value objects, are extracted as the associated context elements because many embedded systems are abstracted as a sensor-and-actuator machine exchanging the physical value between a system and its context. Applying the *Guide Words for Context Analysis*, we can explore only a sequence of context elements directly or indirectly affecting the data value observed or controlled by the system sensors and actuators. Other context elements not affecting the system observation and control are not taken into account because these context elements do not affect the system behavior. We can deal with the *frame problem* because we only have to consider limited number of context elements as the context of the target system.

The remainder of this paper is structured as follows. In Section 2, problems in the current requirements analysis methods are pointed out in terms of the *frame problem*. In Section 3 and 4, CAMEmb is introduced to deal with the *frame problem*. In Section 5, we discuss on the relation between CAMEmb and the *problem frame approach* [5]. Moreover, we discuss how to apply our idea to other domains such as security. Concluding remarks are provided in Section 6.

2 Motivation

In this section, typical problems in the current requirements analysis methods are pointed out by describing the specification of an electric pot as an example.

2.1 Motivating Example

An electric pot is an embedded system for boiling water. Here, for simplicity, only the following is considered: 1) the pot has three hardware components: a heater, a thermostat, and a water level sensor; 2) the pot controls the water temperature by turning on or off the heater; 3) the pot changes its mode from the heating mode to the retaining mode when the temperature becomes 100 Celsius; and 4) the pot observes the volume from the water level sensor that detects whether water is below or above a certain base level.

In case of the electric pot, the water temperature should be taken into account as an important context element. Here, as an example, let us consider the specification that controls the water temperature. In most cases, this specification is described by implicitly taking into account the specific context—for example, such the context that water is boiled under the normal air pressure. A developer describes the software logic corresponding to the specific context—in this case, the pot continues to turn on a heater switch until the water temperature becomes 100 Celsius. Below is the specification described in pseudo code. This function describes that a controller continues to turn on a heater while the value of the temperature obtained from a thermostat is below 100 Celsius. The `Boil` function behaves correctly under the normal circumstance.

```
// Boil function
while thermostat.GetTemperature() < 100.0
  do heater.On();
```

Although this traditional approach is effective, there is room for improvements because it does not explicitly consider the context elements such as water and air pressure. The above `Boil` specification looks correct. However, faults may occur if the expected context is changed—for example, the circumstance of the low air pressure. Because the boiling point is below 100 Celsius under this circumstance, the software controller continues to heat water even if its temperature becomes the boiling point. As a result, water evaporates and finally its volume will be empty. The water level sensor observes the volume, and the pot stops heating. Although this behavior satisfies the above system specification, the pot may be useless for the people who use it up on high mountains where the air pressure is low.

2.2 Problems to be tackled

The boundary of the context should be determined from stakeholders' requirements. If we consider climbers as customers of the pot, we have to admit that we failed in eliciting requirements in the above example.

It is not easy to define the context boundary even if the target users of the system are determined. A developer will be faced with the *frame problem* because there are unlimited context elements in the real world. There are some studies that take into account the real world as a modeling target. For example, Greenspan, S. et al. claim the necessity of introducing real world knowledge into requirement specifications [2]. But, current requirements elicitation methods do not answer a question: how and why do we find air pressure as a context element ? Of course, domain knowledge and past experiences are important to find this kind of requirements elicitation. Moreover, we admit that there are no complete methods to overcome the *frame problem*. However, at the same time, we need a method for systematically exploring the context boundary because many incidents that occur in the real embedded systems are caused by insufficient context analysis. That is, unexpected context influence that cannot be predicted in the requirements elicitation phase tends to cause a crucial incident. Many engineers in the industry face this problem.

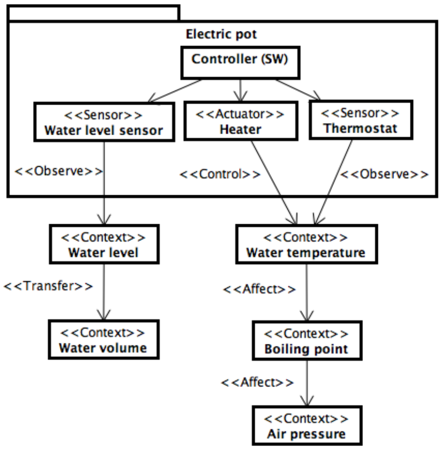


Fig. 1. Context analysis model for an electric pot

3 CAMEmb

CAMEmb is a context analysis method for dealing with the problem pointed out in Section 2. CAMEmb complements the insufficiency of the traditional requirements analysis methods.

3.1 Context analysis model

Figure 1 illustrates the result of context analysis for an electric pot. The upper side and the lower side show a system and its context, respectively. The details of the *Controller* in the context model are described in the system analysis model. Sensors and actuators for observing or controlling the context are regarded as the interface components that separate the context from a system. Figure 1 shows only the structural aspect of the context modeling. The details of the *Controller* and the behavioral aspect of the context model are omitted due to the space limitation. In CAMEmb, the behavioral aspect is modeled using state machine diagrams. The structural aspect plays an important role in exploring the context boundary as mentioned below.

3.2 UML profile for context analysis

A UML profile is provided for context analysis. This profile can describe system elements, context elements, and associations between them: four kinds of stereotypes including $\ll Context \gg$, $\ll Hardware \gg$, $\ll Sensor \gg$, and $\ll Actuator \gg$ are defined as an extension of the UML class ($\ll Sensor \gg$ and $\ll Actuator \gg$ are subtypes of $\ll Hardware \gg$); and five kinds of stereotypes including $\ll Observe \gg$, $\ll Control \gg$, $\ll Transfer \gg$, $\ll Affect \gg$, and

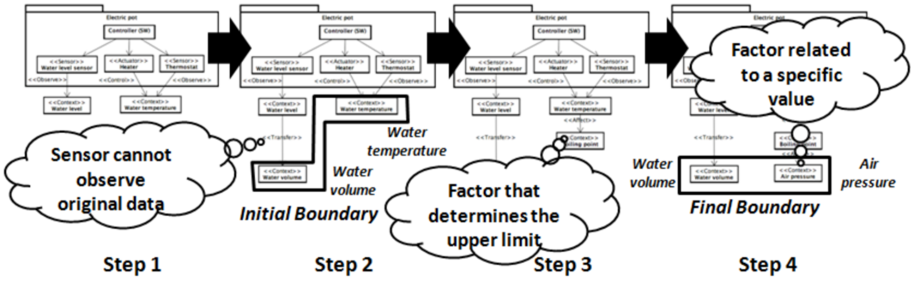


Fig. 2. Stepwise context analysis using guide words (for illustration only)

≪ $Noise$ ≫ are defined as an extension of the UML association. The arrow of ≪ $Observe$ ≫ and ≪ $Control$ ≫ indicates the target of observation and control. The arrow of ≪ $Noise$ ≫ and ≪ $Affect$ ≫ indicates the source of noise and affect, respectively. The arrow of ≪ $Transfer$ ≫ indicates the source of transformation. The associations between *Controller* and three hardware components (sensors and actuators) indicate the phenomena such as *sending a command from software to hardware* and *receiving data from hardware*. However, stereotypes for these phenomena are not provided in our UML profile because they should be considered in system analysis not in context analysis.

4 Stepwise context analysis using guide words

The context model shown in Figure 1 is created as illustrated in Figure 2. Figure 2 shows only the image of context analysis procedures. Please refer to Figure 1 when a detailed analysis result is needed.

Step1: extract directly observed or controlled context elements

First, context elements (≪ *Context* ≫), which are directly observed or controlled by a sensor or an actuator, are extracted. We regard the environment value as a context element because CAMEmb focuses on embedded systems based on sensing and actuating. We call these context elements *"value-context elements"*. In case of an electric pot, *water level* and *water temperature* are extracted since *water level* is observed by the water level sensor and *water temperature* is controlled by the heater.

Step 2 [Initial boundary]: extract indirectly observed or controlled context elements

An element directly observed by a sensor may be an alternative context element in such a case that the sensor cannot observe the original value of the target context element. For example, the pot wants to observe not the *water level*

Table 1. Guide words for context analysis

No.	Category of $\ll Affect \gg$	Guide word
1.	physical phenomena	factor that determines the upper limit
2.	physical phenomena	factor that determines the lower limit
3.	physical phenomena	factor related to a specific value
4.	influence to sensing	factor that interferes with the observation
5.	influence to actuation	factor that interferes with the control

but the *water volume*. Next, we explore the target context elements by using $\ll Transfer \gg$. In the step 2, all paths from sensors and actuators to the target context elements are completely extracted. The initial context boundary is determined in this stage. In case of an electric pot, *water volume* and *water temperature* are extracted as the initial context boundary.

Step 3 [intermediate boundary]: extract impact factors using guide words

The initial context boundary is an ideal boundary in which system's sensing and controlling are not affected by other factors. However, there are many factors affecting observation and actuation in the real world. We have to extract these factors in order to develop reliable embedded systems.

In CAMEmb, impact factors that affect the states of these context elements are extracted using guide words. Guide words, hints for deriving related elements, are effective for software deviation analysis [6]. Guide words are mainly used in HAZOP (Hazard and Operability Studies). In HAZOP, deviation analysis is performed by using the guide words including *NOT*, *MORE*, *LESS*, *AS WELL AS*, *PART OF*, *REVERSE*, and *OTHER THAN*. For example, *higher pressure*, which may be deviated from a normal situation, can be derived from the property *pressure* and the guide word *high*.

In addition to the HAZOP guide words, CAMEmb provides a set of guide words specific to the context analysis as shown in Table 1. These guide words help us to find an obstacle that affects the system observation and control in terms of the *context-value*. By using these guide words, we can extract context elements that affect the context elements existing within the initial boundary. If there is a context element having the influence on another context element, we link them by the $\ll Affect \gg$ association.

In case of an electric pot, the *boiling point* can be extracted as an impact factor for the *water temperature* by applying the guide word *"factor that determines the upper limit"* since the temperature does not become higher than the boiling point. Step 3 in Figure 2 shows this stage of the context analysis.

4.1 Step 4 [Final boundary]: determine the context boundary

We have to continue to extract impact factors as many as possible to develop reliable systems. In case of an electric pot, the *air pressure* can be extracted as an impact factor for the *boiling point* by applying the guide word *"factor related to a specific value"* since the boiling point of the water is 100 Celsius under the circumstance of 1.0 atm. At this point, we finish the context exploration because we can find no more impact factors affecting the *air pressure*.

We can extract two context elements *water volume* and *air pressure* as the final context boundary.

As shown here, the boundary of the context is explored by using *UML Profile for Context Analysis* and *Guide Words for Context Analysis*. We can explore only a sequence of context elements directly or indirectly affecting the data value observed or controlled by the system sensors and actuators. Other context elements not affecting the system observation and control are not extracted. There are many context elements such as person, table, and light in the environment of an electric pot. However, these context elements do not affect the data observed or controlled by the pot. So, we do not have to take into account these context elements. These context elements exist out of the boundary.

5 Discussion

5.1 Avoidance of the frame problem

In CAMEmb, we select only the elements affecting the data value observed or controlled by a system. We think that the value-based context analysis is reasonable because most embedded systems observe the input data from the environment through sensors and affect the environment by emitting the physical outputs through actuators. The system behavior is determined by the data observed by the sensors and controlled by the actuators. We have only to take into account the context elements explicitly or implicitly affecting the data linked with the $\ll Transfer \gg$ or the $\ll Affect \gg$ associations. The context analysis terminates when there are no more context elements affecting the data. In our approach, the affection is determined by using guide words. Of course, the method using guide words is not complete. But, the method helps a developer to find the context elements affecting the system behavior as many as possible.

5.2 Problem frames

Jackson, M. proposes the *problem frames approach* in which relations between a machine (a system to be developed) and the real world are explicitly described. The approach emphasises on the importance of analysing the real world and the problems. The notion of context in CAMEmb corresponds to the real world in the problem frame. Examples of formalising requirements with problem frames can be found in [1] [3]. We believe that CAMEmb provides a fruitful mechanism for using the *problem frames approach* more effectively. The *problem frames approach* is strong in analysing the real world (context) in terms of the problems. On the other hand, CAMEmb is strong in exploring the context boundary.

5.3 Application to other domains

Parnas, D. L. and Madey J. propose the *four-variable model* [8] in which the functions, timing, and correctness are described by using monitored variables, control variables, and input / output data items. The *four-variable model* was used to specify the requirements for the A-7 aircraft in SCR (Software Cost Reduction) [4] providing a tabular notation for specifying requirements. The *four-variable model* is similar to CAMEmb because monitored variables and control variables correspond to context elements observed by sensors and controlled by actuators.

Although we may not be able to apply CAMEmb to all the application domains, there are many domains that can be modelled as monitor-controller (or sensor-actuator) systems. Security, safety, network threats, and user interactions are examples of such domains. In these domains, context can be analyzed using our approach. For example, *trust* in the security domains correspond to *value* in CAMEmb. By defining the *guide words* that affect the trusts, we can explore the trust boundary.

6 Conclusion

In this paper, we proposed CAMEmb, a context-dependent requirements analysis method. As demonstrated in this paper, we could provide a method for exploring the context boundary. The idea of *value-context elements* and *guide words* plays an important role. We think that the essential idea of CAMEmb can be applied to other kinds of context such as security and safety in embedded systems. As the next step, we plan to apply CAMEmb to such an application.

References

1. Coleman, J. W. and Jones, C. B.: Examples of how to Determine the Specifications of Control Systems, In *Proceedings of Workshop on Rigorous Engineering of Fault-Tolerant Systems (REFT 2005)*, pp.65-73, 2005.
2. Greenspan, S., Mylopoulos, J., and Borgida, A.: Capturing More World Knowledge in the Requirements Specification, In *Proceedings of International Conference on Software Engineering (ICSE'82)*, pp.225-234, 1982.
3. Hayes, I., Jackson, M., and Jones, C.: Determining the specification of a control system from that of its environment, In *International Symposium for Formal Methods Europe (FME 2003)*, pp.154-169, 2003.
4. Heitmeyer, C. L., Bull, A., Gasarch, C., and Labaw, B. G. SCR*: A Toolset for Specifying and Analyzing Requirements, In *Proceedings of Computer Assurance (COMPASS)*, pp.109122, 1995.
5. Jackson, M: *Problem Frame: Analyzing and Structuring Software Development Problems*, Addison-Wesley, 2001.
6. Leveson, N. G.: *Safeware: System Safety and Computers*, Addison-Wesley Publishing Company, 1995
7. McCarthy, J. and Hayes, P. J.: Some Philosophical Problems from the Standpoint of Artificial Intelligence, *Machine Intelligence*, 4, pp.463-502, 1969.
8. Parnas, D. L. and Madey, J.: Functional Documentation for Computer Systems Engineering, *McMaster University, Technical Report CRL 237*, 1991.

OLAP Visualization Operator for Complex Data

Sabine Loudcher and Omar Boussaid

ERIC laboratory, University of Lyon (University Lyon 2)
5 avenue Pierre Mendes-France, 69676 Bron Cedex, France
Tel.: +33-4-78772320, Fax: +33-4-78772375
(omar.boussaid, sabine.loudcher)@univ-lyon2.fr

Abstract. Data warehouses and Online Analysis Processing (OLAP) have acknowledged and efficient solutions for helping in the decision-making process. Through OLAP operators, online analysis enables the decision-maker to navigate and view data represented in a multi-dimensional manner. But when the data or objects to be analyzed are complex, it is necessary to redefine and enhance the abilities of the OLAP. In this paper, we suggest combining OLAP and data mining in order to create a new visualization operator for complex data or objects. This operator uses the correspondence analysis method and we call it VOCoDa (Visualization Operator for Complex Data).

Keywords : OLAP, Data Mining, Complex Data, Visualization

1 Introduction

Data warehouses and Online Analysis Processing (OLAP) have recognized and effective solutions for helping in the decision-making process. Online analysis, thanks to operators, makes it possible to display data in a multi-dimensional manner. This technology is well-suited when data are simple and when the facts are analyzed with numeric measures and qualitative descriptors in dimensions. However, the advent of complex data has questioned this process of data warehousing and online analysis.

Complex data often contain a document, an image, a video, ..., and each of these elements can be described and observed by a set of low-level descriptors or by semantic descriptors. This set of elements can be seen not only as complex data but also as a complex object. A complex object is a heterogeneous set of data, which, when combined, form a semantic unit. For instance, a patient's medical record may be composed by heterogeneous elements (medical test results, X-rays, ultrasounds, medical past history, letter from the current doctor, ...) and is a semantic unit. It is a complex object.

As said above, warehousing and online analytical processes must be modified in the case of complex objects. In this paper, we focus on the visualization of complex objects. The problem of storing and modeling complex objects is discussed in other articles. The purpose of online analysis is to (1) aggregate many data to summarize the information they contain; (2) display the information according to different dimensions (3) navigate through data to explore them. OLAP operators are well-defined for classic data. But they are inadequate when data are complex. The use of other techniques, for example data mining, may be promising. Combining data mining methods with OLAP tools is an interesting solution for enhancing the ability of OLAP to analyze complex objects. We have already suggested extending OLAP capabilities with complex object exploration and clustering.

In this paper, we are concerned with the problem of the visualization of complex objects in an OLAP cube. By this means, we aim to define a new approach to extending OLAP capabilities to complex objects. With the same idea of combining data mining and online analysis, some works suggest using *Visual Data Mining* technology for visually and interactively exploring OLAP cubes. Maniatis *et al.* list possible representations for displaying a cube and offer the CPM model (*Cube Presentation Model*) as a model in an OLAP interface [3]. The CPM model borrows visualization tools from the field of the HMI (Human Machine Interface). Unfortunately, these works do not take complex objects into account . In a cube of complex objects, the facts are indeed complex objects, and the dimensions can include images, texts, descriptors, ... and OLAP measures are not necessarily numeric. Given these characteristics, standard visualization tools are not necessarily well-suited and should be adapted. To do this, we use the well-known principle of the factor analysis method in data mining. Factor analysis makes it possible to visualize complex objects while highlighting interesting aspects for analysis. This technique represents objects by projecting them on to factor axes. In a previous paper, we laid the foundations for this proposal [4]. In this paper, we complete and improve our first proposal by taking into account the measure to visualize complex objects, using indicators to make interpretation easier. We thus offer a comprehensive approach and a new OLAP operator entitled VOCoDa (*Visualization Operator for Complex Data*).

2 Running example

To illustrate our point of view, we complete the previously used case of researchers' publications. A publication can be seen as a complex object, or as a semantic entity. We plan to analyze publications according to their authors, national or international range, support such as a conference or a journal, etc. We aim to observe the diversity of the themes in which researchers publish and the proximity of authors when they are working on the same themes. Here, we observe publications as complex objects. To handle these semantic entities, we therefore need an adapted modeling and analysis tools.

In addition to standard descriptors such as year, type, authors, number of pages, etc., the user may also want to analyze the semantic content of the publication, i.e. the topics of the publication. The semantic content of the publication must be taken into account when modeling and carrying out an analysis. Let us suppose that the user wants to analyze publications according to the first author, support, year, content and topics of the paper.

The obtained multidimensional model is shown in figure 1.

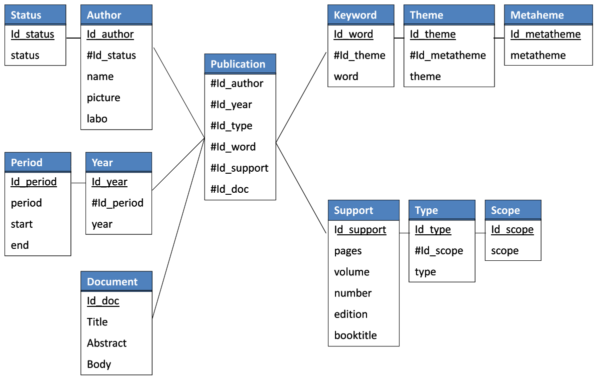


Fig. 1. Multidimensional modeling of publications

In this model, we believe that each dimension can be the fact and that objects are interchangeable in multi-dimensional modeling. There are therefore ”classic” dimensions with hierarchies, and semantic dimensions consisting of a hierarchy of concepts (keywords−− >themes−− >metathemes) and the document itself. Here, the fact is the publication and it is a combination of all dimensions without a measure. Generally, in case like this where there are no measures , the aggregation function COUNT can be used to count the facts. This solution is always possible in our case, but it is not sufficient because the analysis which follows is too poor. We seek other means to analyze publications in order to discover thematic proximity, authors who work together,... We consider a publication as a complex object and we are looking for a way to make a semantic analysis. We propose a visualization of complex objects which takes the semantic content of objects into account. This explains our decision to use a factor analysis method for the visualization of complex objects. This new visualization method fits completely with the online analysis of complex objects.

3 Positioning and principle

Generally, OLAP interfaces represent a cube as a table, or cross-table. In an attempt to exceed the limits of standard interfaces, more advanced tools offer visual alternatives to represent the information contained in a cube, and to interactively browse the cube (hierarchical visualizations, trees of decomposition, multi-scale views, interactive scatter plots) [8]. For a better visualization of information, Sureau *et al.* suggest rearranging the modalities of a level according to heuristics, based on distance between the elements in a dimension or according to a genetic algorithm [7]. With a statistical test, Ordonez and Chen searched within a cube (of low dimension) for neighboring cells with significantly different measures [6]. In the context of Web and OLAP applications, Aouiche *et al.* use a tag cloud to represent a cube where each keyword is a cell and where keyword size depends on the measured value of the fact (cell) [1].

Compared with the other approaches presented, we suggest a visualization operator (1) in the context of online analysis (2) that requires no assumptions about the data (3) that is suitable for complex objects (4) and that takes the semantic content of the data into account. Works on OLAP visualization do not deal with complex objects (even if some might be adapted to such data) and do not take the semantic content (only tag clouds seem to do this) into account.

To visualize complex objects, we propose an approach that uses factor analysis, a well-known method in data mining [2], [5]. A factor method makes it possible to visualize complex objects while highlighting interesting facts for analysis. When facts are complex objects, often there is no measure in the classical sense of multi-dimensional modeling. However, it is always possible to count the facts. In this case, the complex object cube with several dimensions with the COUNT function can be seen as a contingency table. Correspondence analysis (CA) can be used to display the facts. CA produces factor axes which can be used as new dimensions, called "factor dimensions". These new axes or dimensions constitute a new space in which it is possible to plot the facts i.e. complex objects. Using CA as the visualization operator is fully justified because this method has the same goal as OLAP navigation and exploration.

4 Process

We provide OLAP users with a process composed of several steps: (1) building the complex object cube, (2) constructing the contingency table, (3) completing the correspondence analysis, (4) mapping complex objects on the factorial axes.

Suppose that the user wants to study keywords in order to identify the major research fields in which researchers are working. In addition, the user would like to identify researchers working on the same keywords.

4.1 Notations

Let $\mathcal{C}$ be a cube with a non-empty set of d dimensions $\mathcal{D} = \{D^1, ..., D^i, ..., D^d\}$ and m measures $\mathcal{M} = \{M_1, ..., M_q, ..., M_m\}$. $\mathcal{H}^i$ is the set of hierarchical levels of dimension D^i. H^i_j is the j hierarchical level of dimension D^i. For example, the type of publication dimension D^1 has two levels: the level *Type* denoted H^1_1 and the level *Scope* denoted H^1_2.

$\mathcal{A}^{ij} = \{a^{ij}_1, ..., a^{ij}_t, ..., a^{ij}_l\}$ is the set of the l members or modalities a^{ij}_t of the hierarchical level H^i_j of the dimension D^i. The level *Scope* (H^1_2) has two members: *International*, denoted a^{12}_1 and *National*, denoted a^{12}_2.

4.2 Complex object cube

Depending on what the user wants to analyze, a cube is defined. This constructed cube is a sub-cube from the initial cube C. Let $\mathcal{D}'$ be a non-empty sub-set of $\mathcal{D}$ with p dimensions $\{D^1, ..., D^p\}$ ($\mathcal{D}' \subseteq \mathcal{D}$ and $p \leq d$). The p-tuple $(\Theta^1, ..., \Theta^p)$ is sub-cube if $\forall i \in \{1, ..., p\}$, $\Theta^i \neq \emptyset$ and if there is an unique $j \geq 1$ such that $\Theta^i \subseteq \mathcal{A}^{ij}$. A sub-cube, noted $\mathcal{C}'$, corresponds to a portion from the initial cube $\mathcal{C}$. Of the d existing dimensions, only p are chosen. For each chosen dimension $D^i \in \mathcal{D}'$, a hierarchical level H^i_j is selected and a non-empty sub-set Θ^i of members is taken from all the member set $\mathcal{A}^{ij}$ of the level.

For example, the user can choose to work in the context of the publications that were written between 2007 and 2009, by authors with the status of full professor. And in this context, the user can build, a cube of publications based on keywords, year of publication and the name of the first author. In our example, the sub-cube is given by $(\Theta^1, \Theta^2, \Theta^3, \Theta^4)$= ({*full professor*},{*2007, 2008, 2009*},{*Keyword 1, Keyword 2, ..., Keyword 4*},{*Author 1, Author 2, ..., Author 4*}). The measure M_q is the number of publications (*Count*).

4.3 Contingency table

Classically, correspondence analysis takes as input a contingency table. Our idea is to use traditional OLAP operators to build this contingency table.

In the sub-cube $\mathcal{C}'$, the user chooses two levels (one level for two different dimensions), on which he wants to visualize complex objects. Let Θ^i (respectively $\Theta^{i'}$) be the set of l (respectively l') members chosen for the level of the dimension i (respectively i'). The contingency table $\mathcal{T}$ has l rows and l' columns the titles of which are given by $\{a^{ij}_1, ..., a^{ij}_t, ..., a^{ij}_l\}$ and $\{a^{i'j'}_1, ..., a^{i'j'}_{t'}, ..., a^{i'j'}_{l'}\}$. At each intersection of row t and column t', are counted the facts having the members a^{ij}_t and $a^{i'j'}_{t'}$.

In our example, the contingency table crosses keywords with authors in the sub-cube. This consists in counting facts covering 3 years by doing a roll-up of the dimension *year*. This gives us a cross table with keywords in rows and authors in columns. At the intersection of a row and a column, we have the number of publications written by an author for a given keyword. This table is ready to be processed by a CA. If the measure used is other than a simple count, and if it is a numerical measure, additive and with only positive values, then it is possible to use it to weigh the facts in the contingency table. The user is given the choice of using this measure as weighting or not.

4.4 Correspondence analysis

Processing a CA consists in projecting data on to synthetic axes so that much information is expressed by a minimum number of axes. The goal is to reduce the size of the representation space, that is to say, to reduce the number of rows and columns. The CA makes possible simultaneous visualization of the projections of rows and columns in the same plane. The proximities between rows and columns can be interpreted.

In practice, the method starts by calculating the eigen values from which are deduced eigen vectors that define the factor axes. As the first two axes contain the most information, they define the first factor plane. Once row points and column points have been projected on to axes, auxiliary statistics are reported to help evaluate the quality of the axes and their interpretation. For each point, the most important statistics are the weight, the relative contribution of the point to the axis' inertia and the quality of the representation on the axis (given by the $cosine^2$). To give an interpretation of an axis and analyze proximity between points on an axis, only points which contribute strongly to the inertia of the axis (whose contribution is three times the average contribution) and which are well represented by the axis (whose $cosine^2$ is higher than 0.5) are taken into account.

4.5 Visualization

The first two factor axes are retained as new factor dimensions, because the coordinates of the projected objects can be seen as members of dimensions. The graph in figure 2 is obtained. It allows representing publications according to their semantic content described by authors and keywords. It is possible to interpret the factor dimensions. Once the graph has been constructed, an interactive tool gives, for each point, i.e. keyword or author, its statistic indicators (relative contribution and $cosine^2$). Keywords and authors that have high indicators are represented in a different color. Thus, the user sees the most relevant points for analysis. Factor analysis provides automatic help in understanding and to analyzing information. For example, the user can easily identify the most characteristic keywords, authors who work together or who do not work together and finally groups of authors working on certain keywords. In addition, if the user so requests, a photograph of the authors can replace their name. In an OLAP framework, it is efficient to use the most significant descriptors of dimensions in order to enhance the readability of the results obtained.

Furthermore, according to the OLAP principle, it is also possible on each point to perform a *drill-down* to see related publications (represented by their title). The user has another possibility of projecting a hierarchical level of another dimension into the graph. The members of this new level will be projected as points in factor space but they have not been involved in the construction of the axes. To maintain statistical consistency, only hierarchical levels whose dimensions are not in the sub-cube can be used as additional elements. A level of a dimension already used would be dependent on another level. In our example, the user could use as an additional element type of publication (journal, conference, technical report ...).

We have developed a software platform implemented as a Web Open Source application in *PHP5* and with a *MySQL* database. It uses the *R* software and its *FactoMiner* package. The graphic interface is managed by an *ExtJS* framework with an *Ajax* support.

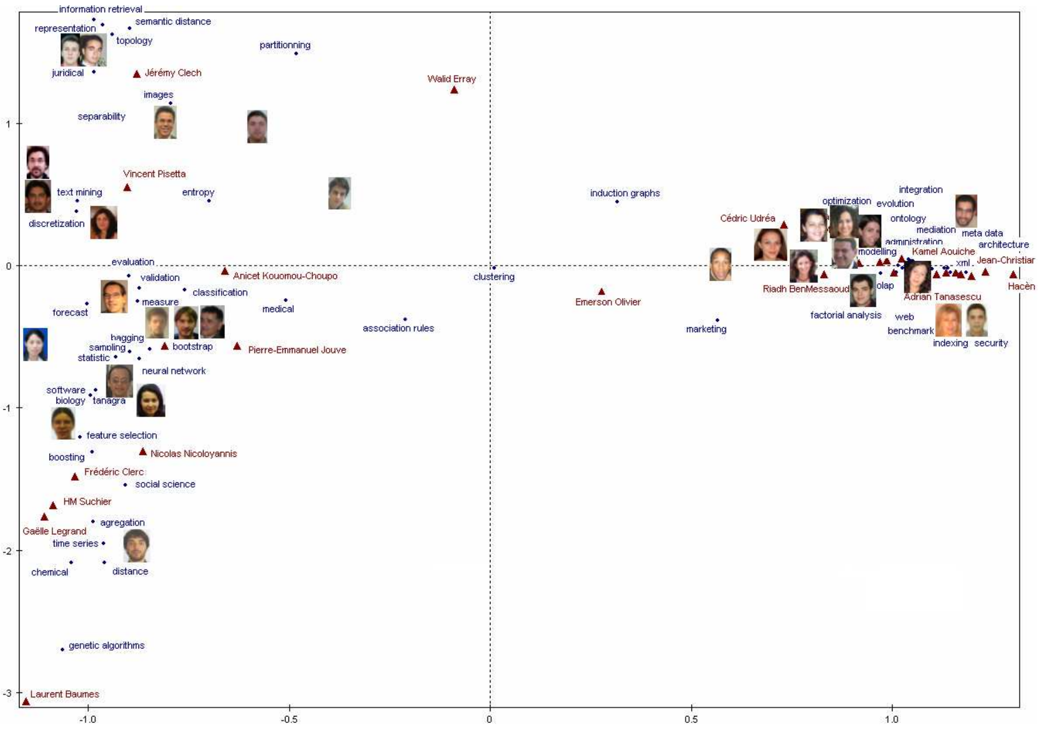


Fig. 2. Visualization of publications

5 Conclusion

In this paper, we have developed an approach to online analysis for complex objects. Our approach has demonstrated the feasibility of using correspondence analysis to make it possible to visualize complex objects online taking their semantic content into account. Furthermore, it naturally takes its place in the online analysis. The publications case study illustrates our approach. In the proposed multi-dimensional model, publications are described by keywords. Rather than asking authors to assign keywords themselves manually to their publication or rather than using an ontology, we think that it would be more relevant to automatically extract the keywords from the title, summary, or text (body) of the publication. Indeed, if the keywords were automatically extracted, they would capture some of the semantics contained in the document. Using information retrieval (IR) principles, keywords could be extracted automatically. Furthermore, as publications contain documents and documents contain text, our idea is to use certain information retrieval (IR) techniques in order to model publications. The use of IR techniques can allow us to extract semantics from the text and this semantic information may be very helpful for modeling publications in a multi-dimensional manner. In addition to combining OLAP and data mining, the coupling of OLAP and IR should further enhance online analysis.

References

1. K. Aouiche, D. Lemire and R. Godin. Web 2.0 OLAP: From Data Cubes to Tag Clouds. Proceedings of the 4^{th} International Conference on Web Information Systems and Technologies (WEBIST 08). 2008, 5–12.
2. J.P. Benzecri. Correspondence Analysis Handbook. Marcel Dekker, hardcover edition, 1992.
3. A. S. Maniatis, P. Vassiliadis, S. Skiadopoulos, Y. Vassiliou. Advanced visualization for OLAP. Proceedings of the 6^{th} ACM International Workshop on Data Warehousing and OLAP (DOLAP'2003). 2003,9–16.
4. L. Mabit, S. Loudcher, O. Boussaid. Analyse en ligne d'objets complexes avec l'analyse factorielle. 10^{me} Confrence d'Extraction et Gestion des Connaissances (EGC 2010). 2010, 381–386.
5. M. Greenacre. Correspondence Analysis in Practice. Chapman Hall CRC, Second Edition. 2007.
6. C. Ordonez, Z. Chen. Exploration and Visualization of OLAP Cubes with Statistical Tests. Proceedings of the 15^{th} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Workshop on Visual Analytics and Knowledge Discovery. 2009,46–55.
7. F. Sureau, F. Bouali, G. Venturini. Optimisation heuristique et gntique de visualisations 2D et 3D dans OLAP : premiers rsultats. 5mes Journes francophones sur les Entrepts de Donnes et l'Analyse en ligne (EDA 09). 2009, 62–75.
8. S. Vinnik, F. Mansmann. From analysis to interactive exploration: Building visual hierarchies from OLAP cubes. Proceedings of the 14^{th} International Conference on Extending Database Technology (EDBT'2006). 2006, 496–514.

When Process Mining Meets Bioinformatics

R.P. Jagadeesh Chandra Bose[1,2] and Wil M.P. van der Aalst[1]

[1] Department of Mathematics and Computer Science, University of Technology,
Eindhoven, The Netherlands
[2] Philips Healthcare, Veenpluis 5–6, Best, The Netherlands
{j.c.b.rantham.prabhakara,w.m.p.v.d.aalst}@tue.nl

Abstract. Process mining techniques can be used to extract non-trivial process related knowledge and thus generate interesting insights from event logs. Similarly, bioinformatics aims at increasing the understanding of biological processes through the analysis of information associated with biological molecules. Techniques developed in both disciplines can benefit from one another, e.g., sequence analysis is a fundamental aspect in both process mining and bioinformatics. In this paper, we draw a parallel between bioinformatics and process mining. In particular, we present some initial success stories that demonstrate that the emerging process mining discipline can benefit from techniques developed for bioinformatics.

Keywords: sequence, trace, execution patterns, diagnostics, conformance, alignment, configuration

1 Introduction

Bioinformatics aims at increasing the understanding of biological processes and entails the application of computational techniques to understand and organize the information associated with biological macromolecules [1]. Sequence analysis or sequence informatics is a core aspect of bioinformatics that is concerned with the analysis of DNA/protein sequences[3] and has been an active area of research for over four decades.

Process mining is a relatively young research discipline aimed at discovering, monitoring and improving real processes by extracting knowledge from event logs readily available in today's information systems [2]. Business processes leave trails in a variety of data sources (e.g., audit trails, databases, transaction logs). Hence, every process instance can be described by a trace, i.e., a sequence of events. Process mining techniques are able to extract knowledge from such traces and provide a welcome extension to the repertoire of business process analysis techniques. The topics in process mining can be broadly classified into three

[3] DNA stores information in the form of the base nucleotide sequence, which is a string of four letters (A, T, G and C) while protein sequences are sequences defined over twenty amino acids and are the fundamental determinants of biological structure and function.

categories (i) *discovery*, (ii) *conformance*, and (iii) *enhancement*. Process discovery deals with the discovery of models from event logs. For example, there are dozens of techniques that automatically construct process models (e.g., Petri nets or BPMN models) from event logs [2]. Discovery is not restricted to control-flow; one may also discover organizational models, etc. Conformance deals with comparing an apriori model with the observed behavior as recorded in the log and aims at detecting inconsistencies/deviations between a process model and its corresponding execution log. In other words, it checks for any violation between *what was expected to happen* and *what actually happened*. Enhancement deals with extending or improving an existing model based on information about the process execution in an event log. For example, annotating a process model with performance data to show bottlenecks, throughput times etc. Some of the challenges in process mining include the discovery of process maps (navigable hierarchical process models) and the provision of process diagnostics support for auditors and analysts [3].

It is important to note that, to a large extent, sequence analysis is a fundamental aspect in almost all facets of process mining and bioinformatics. In spite of all the peculiarities specific to business processes and process mining, the relatively young field of process mining should, in our view, take account of the conceptual foundations, practical experiences, and analysis tools developed by sequence informatics researchers over the last couple of decades. In this paper, we describe some of the analogies between problems studied in both disciplines. We present some initial successes which demonstrate that process mining techniques can benefit from such a cross-fertilization.

2 Notations

We use the following notations in this paper.

- Let Σ denote the set of activities. Σ^+ is the set of all non-empty finite sequences of activities from Σ.
- A trace corresponds to a process instance expressed as a finite sequence of activities. $T \in \Sigma^+$ is a trace over Σ. $|T|$ denotes the length of the trace T.
- The ordered sequence of activities in T is denoted as $T(1)T(2)T(3)\ldots T(n)$ where $T(k)$ represents the k^{th} activity in the trace.
- An event log, $\mathcal{L}$, corresponds to a multi-set (or bag) of traces from Σ^+.

3 From Sequence to Structure

A DNA *sequence motif* is defined as a nucleic acid *sequence pattern* that has some biological significance (both structural and functional) [4]. These motifs are usually found to recur in different genes or within a single gene. For example, *tandem repeats* (tandemly repeating DNA) are associated with various regulatory mechanisms such as protein binding [5]. More often than not, sequence motifs are also associated with *structural motifs* found in proteins thus establishing a strong correspondence between sequence and structure.

Likewise, common subsequences of activities in an event log that are found to recur within a process instance or across process instances have some domain (functional) significance. In [6], we adopted the sequence patterns (e.g., tandem repeats, maximal repeats etc.) proposed in the bioinformatics literature, correlated them to commonly used process model constructs (e.g., tandem repeats and tandem arrays correspond to simple loop constructs) and proposed a means to form abstractions over these patterns. Using these abstractions as a basis, we proposed a *two-phase approach to process discovery* [7]. The first phase comprises of pre-processing the event log with abstractions at a desired level of granularity and the second phase deals with discovering the *process maps* with seamless zoom-in/out facility. Figure 1 summarizes the overall approach.

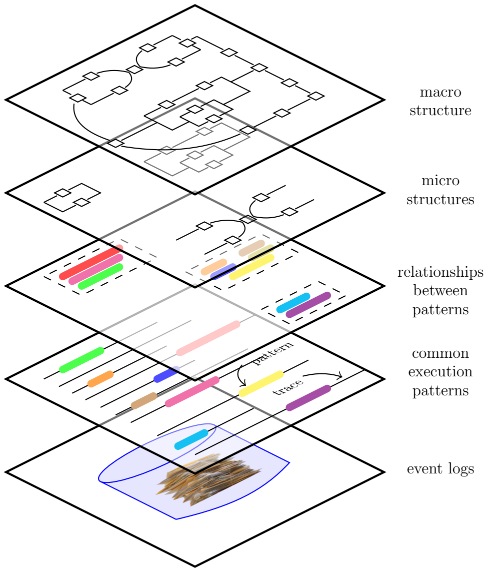


Fig. 1. Repeating subsequences of activities define the common execution patterns and carry some domain (functional) significance. Related patterns and activities pertaining to these patterns define abstractions that correspond to micro-structures (or sub-processes). The top-level process model can be viewed as a macro-structure that subsumes the micro-structures.

Figure 2 highlights the difference between the traditional approach to process discovery and the two-phase approach. Note that the process model (map) discovered using the two-phase approach is simpler. Our approach supports the abstraction of activities based on their context and type, and provides a seamless zoom-in and zoom-out functionality.

Thus the bringing together of concepts in bioinformatics to process mining has enabled the discovery of hierarchical process models and opened a new perspective in dealing with fine granular event logs.

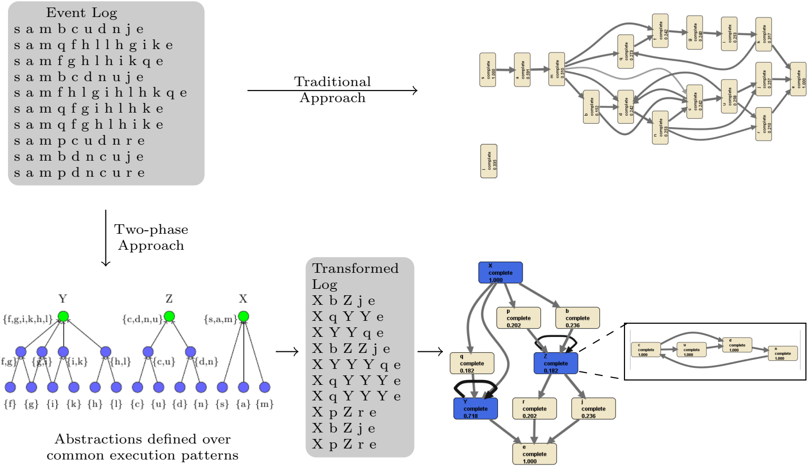


Fig. 2. Traditional approach vs. our two-phase approach to process discovery

4 Sequence Alignment and Process Diagnostics

Multiple sequence alignment has been a subject of extensive research in computational biology for over three decades. Sequence alignment is an essential tool in bioinformatics that assists in unraveling the secondary and tertiary structures of proteins and molecules, their evolution and functions, and in inferring the taxonomic, phylogenetic or cladistic relationships between organisms, diagnoses of genetic diseases etc [8, 9].

In [10], we have adapted sequence alignment to traces in an event log and showed that it carries significant promise in process diagnostics. The goal of *trace alignment* is to align traces in such a way that event logs can be easily explored. Given a set of traces $\mathbb{T} = \{T_1, T_2, \dots, T_n\}$, trace alignment can be defined as a mapping of $\mathbb{T}$ to another set of traces $\overline{\mathbb{T}} = \{\overline{T_1}, \overline{T_2}, \dots, \overline{T_n}\}$ where $\overline{T_i} \in (\Sigma \cup \{-\})^+$ for $1 \leq i \leq n$. In addition, the following three properties need to be satisfied with respect to $\mathbb{T}$ and $\overline{\mathbb{T}}$: (a) each trace in $\overline{\mathbb{T}}$ is of the same length i.e., there exists an $m \in \mathbb{N}$ such that $|\overline{T_1}| = |\overline{T_2}| = \cdots = |\overline{T_n}| = m$ (b) $\overline{T_i}$ is equal to T_i after removing all gap symbols '$-$' and (c) there is no $k \in \{1, \dots, m\}$ such that $\forall_{1 \leq i \leq n}\ \overline{T_i}(k) = -$.

Trace alignment can be used to explore the process in the early stages of analysis and to answer specific questions in later stages of analysis. More specifically, trace alignment can assist in answering questions such as:

- What is the most common (likely) process behavior that is executed?
- Where do my process instances deviate and what do they have in common?
- Are there any common patterns of execution in my traces?
- What are the contexts in which an activity or a set of activities is executed in my event log?

- What are the process instances that share/capture a desired behavior either exactly or approximately?
- Are there particular patterns (e.g., milestones, concurrent activities etc.) in my process?

Figure 3 depicts the results of trace alignment for a real-life log from a rental agency. The figure shows that trace alignment can assist in answering a variety of diagnostic questions. Every row corresponds to a process instance and time increases from left to right. The horizonal position is based on *logical time* rather than real timestamps. If two rows have the same activity name in the same column, then the corresponding two events are very similar and are therefore aligned. Note that the same activity can appear in multiple columns. By reading a row from left to right, we can see the sequence of activities (i.e., the trace) that was executed for a process instance. Process instances having the same trace can be grouped into one row to simplify the diagram. The challenge is to find an alignment that is as simple and informative as possible. For example, the number of columns and gaps should be minimized while having as much consensus as possible per column.

The application of sequence alignment in bioinformatics to process mining has created an altogether new dimension to conformance checking; *deviations and violations are uncovered by analyzing just the raw event traces* (thereby avoiding the need for process models).

Finding good quality alignments is notoriously complex. The initial results of trace alignment are definitely encouraging. Nonetheless, there are various new challenges when adopting biological sequence alignment to trace alignment in the context of business processes [11]. For example, biological sequences tend to be homogenous whereas traces in semi-structured processes (e.g., care processes in hospitals) tend to be much more variable. Other differences are the fact that traces in an event log can be of very different lengths (e.g., due to loops) and may be the result of concurrency. These characteristics provide new challenges for sequence alignment.

5 Phylogeny and Process Configuration

Phylogenetics refers to the study of evolutionary relationships, and is one of the first applications in bioinformatics. A phylogeny is a tree representation of the evolutionary history of a set (family) of organisms, gene/protein sequences etc. The basic premise in phylogenetics is that genes have evolved by duplication and divergence from common ancestors [12]. The genes can therefore exist in a nested hierarchy of relatedness.

In the past couple of years, *process configuration* has gained prominence in the BPM community [13]. Process configuration is primarily concerned with managing families of business processes that are similar to one another in many ways yet differing in some other ways. For example, processes within different municipalities are very similar in many aspects and differ in some other aspects. Such discrepancies can arise due to characteristics peculiar to each municipality

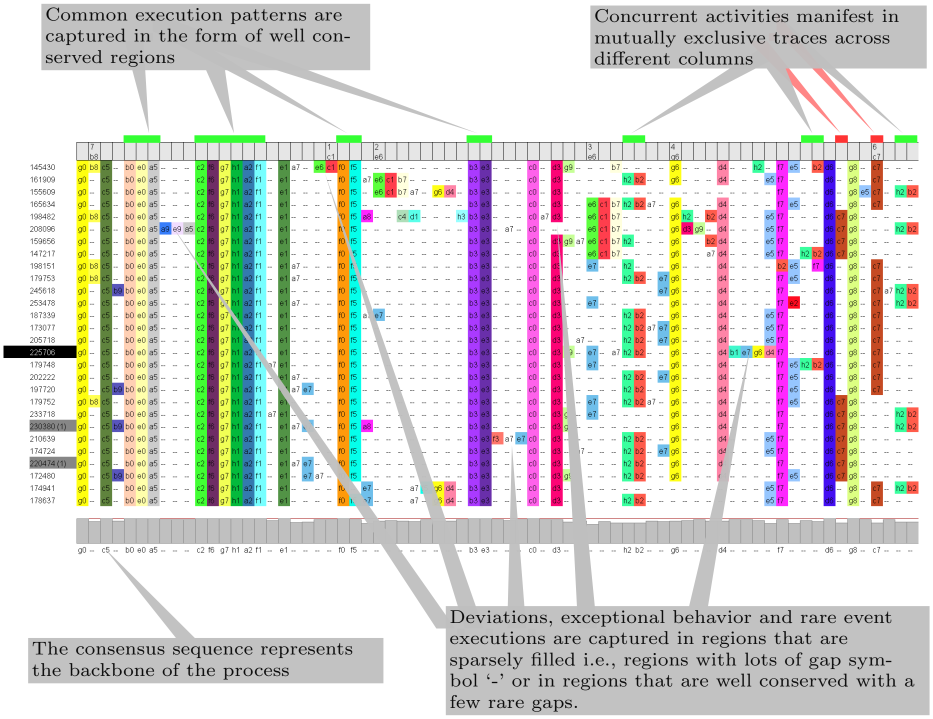

Common execution patterns are captured in the form of well conserved regions
Concurrent activities manifest in mutually exclusive traces across different columns
The consensus sequence represents the backbone of the process
Deviations, exceptional behavior and rare event executions are captured in regions that are sparsely filled i.e., regions with lots of gap symbol '-' or in regions that are well conserved with a few rare gaps.

Fig. 3. An example of trace alignment for a real-life log from a rental agency. Each row refers to a process instance. Columns describe positions in traces. Consider now the cell in row y and column x. If the cell contains an activity name `a`, then `a` occurred for case y at position x. If the cell contains no activity name (i.e., a gap "–"), then nothing happened for y at position x.

(e.g., differences in size, demographics, problems, and policies) that need to be maintained. Furthermore, operational processes need to change to adapt to changing circumstances, e.g., new legislation, extreme variations in supply and demand, seasonal effects, etc. A configurable process model describes a family of similar process models in a given domain [13], and can be thought of as the genesis (root) of the family. All variants in the family can be derived from the configurable model through a series of change patterns [14]. One of the core research problems in *process configuration* is to automatically derive configurable process models from specific models and event logs.

One can find stark similarity between phylogenetics and process configuration. Techniques have been proposed in the bioinformatics literature to discover phylogenies both from (protein) structure as well as from sequences. This can be compared to deriving configurable process models from specific models and from event logs respectively. The adaptability of phylogeny construction techniques to process configuration needs to be explored.

Techniques from bioinformatics have also been adopted to trace clustering in process mining [15, 16]. Sequence clustering techniques have been applied to deal with unlabeled event logs[4] in process mining [17]. Experiences from bioinformatics can also contribute to tooling and infrastructure efforts in process mining. For example, visualization is one of the challenging problems in process mining tooling[5]. A lot of current visualization means in process mining become unmanageable when dealing with large event logs thereby compromising the comprehensibility. *Visualization* is used in many areas within bioinformatics (e.g., sequence matching, genome browsing, multiple sequence alignment etc.), with varying success, and good tools already exist. As another example, to cater to the rapidly increasing accumulation of biological data, lots of efforts had been initiated in bioinformatics to create advanced databases with analysis capabilities devoted to particular categories e.g., Genbank (cataloguing DNA data), SWISS-PROT/TrEMBL (repository of protein sequences) etc. Recently, similar efforts had been initiated in the process modeling and process mining community to create repositories with advanced support for dealing with process model collections e.g., APROMORE [18]. Such an overlap between the goals combined with the promising initial results calls for a more rigorous attempt at understanding and exploiting the synergy between these two disciplines.

6 Conclusions

Bioinformatics and process mining share some common goals. In this paper, we presented the commonalities between the problems and techniques studied in bioinformatics and process mining. Exploiting these commonalities, we demonstrated that process mining can benefit from the plethora of techniques developed in bioinformatics. Initial attempts at such a crossover have enabled the discovery of hierarchical process models and helped extending the scope of conformance checking to also cover the direct inspection of traces. Although this is just a first step towards an interaction between the two disciplines, the results are very promising and the relationship will be explored further in our future work.

Acknowledgments The authors are grateful to Philips Healthcare for funding the research in process mining.

References

1. Luscombe, N., Greenbaum, D., Gerstein, M.: What is Bioinformatics? A Proposed Definition and Overview of the Field. Methods of Information in Medicine **40**(4) (2001) 346–358

[4] In an unlabeled event log, the case to which an event belongs to is unknown.

[5] ProM is an extensible framework that provides a comprehensive set of tools/plugins for the discovery and analysis of process models from event logs. See http://www.processmining.org for more information and to download ProM.

2. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer (2011)
3. van der Aalst, W.M.P.: Challenges in Business Process Mining. Technical Report BPM-10-01, Business Process Management (BPM) Center (2010)
4. Das, M.K., Dai, H.K.: A Survey of DNA Motif Finding Algorithms. BMC Bioinformatics **8**(Suppl 7) (2007) S21
5. Kolpakov, R., Bana, G., Kucherov, G.: mreps: Efficient and Flexible Detection of Tandem Repeats in DNA. Nucleic Acids Research **31**(13) (2003) 3672–3678
6. Bose, R.P.J.C., van der Aalst, W.M.P.: Abstractions in Process Mining: A Taxonomy of Patterns. In Dayal, U., Eder, J., Koehler, J., Reijers, H., eds.: Business Process Management. Volume 5701 of LNCS., Springer-Verlag (2009) 159–175
7. Li, J., Bose, R.P.J.C., van der Aalst, W.M.P.: Mining Context-Dependent and Interactive Business Process Maps using Execution Patterns. In zur Muehlen, M., Su, J., eds.: BPM 2010 Workshops. Volume 66 of LNBIP., Springer-Verlag (2011) 109–121
8. Chan, S., Wong, A.K.C., Chiu, D.: A Survey of Multiple Sequence Comparison Methods. Bulletin of Mathematical Biology **54**(4) (1992) 563–598
9. Gotoh, O.: Multiple Sequence Alignment: Algorithms and Applications. Advanced Biophysics **36** (1999) 159–206
10. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace Alignment in Process Mining: Opportunities for Process Diagnostics. In Hull, R., Mendling, J., Tai, S., eds.: Proceedings of the 8th International Conference on Business Process Management (BPM). Volume 6336 of LNCS., Springer-Verlag (2010) 227–242
11. Notredame, C.: Recent Progress in Multiple Sequence Alignment: A Survey. Pharmacogenomics **3** (2002) 131–144
12. Thornton, J.W., DeSalle, R.: Gene Family Evolution and Homology: Genomics Meets Phylogenetics. Annual Review of Genomics and Human Genetics **1**(1) (2000) 41–73
13. van der Aalst, W.M.P., Lohmann, N., Rosa, M.L., Xu, J.: Correctness Ensuring Process Configuration: An Approach Based on Partner Synthesis. In Hull, R., Mendling, J., Tai, S., eds.: Proceedings of the 8th International Conference on Business Process Management (BPM). Volume 6336 of LNCS., Springer-Verlag (2010) 95–111
14. Weber, B., Rinderle, S., Reichert, M.: Change Patterns and Change Support Features in Process-Aware Information Systems. In: Proceedings of the 19th International Conference on Advanced Information Systems Engineering (CAiSE), Springer-Verlag (2007) 574–588
15. Bose, R.P.J.C., van der Aalst, W.M.P.: Context Aware Trace Clustering: Towards Improving Process Mining Results. In: Proceedings of the SIAM International Conference on Data Mining (SDM). (2009) 401–412
16. Bose, R.P.J.C., van der Aalst, W.M.P.: Trace Clustering Based on Conserved Patterns: Towards Achieving Better Process Models. In: Business Process Management Workshops. Volume 43 of LNBIP., Springer (2010) 170–181
17. Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching Process Mining with Sequence Clustering: Experiments and Findings. In: Proceedings of the 5th International Conference on Business Process Management (BPM). Volume 4714 of LNCS., Springer (2007) 360–374
18. Rosa, M.L., Reijers, H.A., van der Aalst, W.M.P., Dijkman, R.M., Mendling, J., Dumas, M., Garcia-Banuelos, L.: APROMORE: An Advanced Process Model Repository. Expert Systems with Applications **38**(6) (2011) 7029–7040

Towards the Strategic Analysis of Agile Practices

Hesam Chiniforooshan Esfahani[1], Eric Yu[2], Maria Carmela Annosi[3]

[1]Department of Computer Science, University of Toronto
[2]Faculty of Information, University of Toronto
[3]Ericsson Software research, Ericsson Telecommunication, Italy
[1]hesam@cs.toronto.edu, [2]eric.yu@utoronto.ca, [3]mariacarmela.annosi@ericsson.com

Abstract. Agile methods are widely believed to have the potential to improve software processes. Given the variety of agile practices, organizations face difficult decisions on which ones to adopt. Recognizing that agile adoption is often motivated by strategic concerns such as market competitiveness or responsiveness to customer needs, this paper outlines a framework for the strategic analysis of agile practices. The framework aims to support the decision making process leading to agile adoption. The framework builds upon a knowledge base of experiences collected from empirical studies. Goal modeling techniques from requirements engineering are incorporated in the form of a Strategies Graph. The graph resembles the Strategy Map from Balanced Scorecards familiar to many managers.

Keywords. Agile Methods, Situational Method Engineering, Software Process Improvement, Goal-Oriented Modeling, Strategic Management

1 Introduction

Many organizations are changing their software development processes to Agile. A number of frameworks have been proposed to provide guidance for transitioning to agile [1-3], but none takes a strategic perspective to link business goals to the selection of agile practices. This paper introduces the SAAP (Strategic Analysis for Agile Practices) framework for analyzing a set of candidate agile practices from the strategic perspective of an organization. By performing this analysis before enacting any new practices, one can anticipate potential mismatches between organizational strategies and candidate practices.

The analysis procedures of SAAP are mostly focused on agile practices. The framework considers agile methods (either known methods such as XP and Scrum, or those which are custom-built) to be decomposable into agile practices, such as Pair Programming and Daily Meeting. The SAAP framework extends Situational Method Engineering [4], by taking into account organizational strategies as significant situational attributes, which affect the choice of method fragments. The framework takes advantage of a knowledge base of agile practices, containing experiences collected from empirical studies. The knowledge base [5] is created by systematic review of empirical studies which report on the outcomes of different agile practices in various project situations.

The proposed framework consists of three main components: *the Strategies Graph*, the *Evidential Knowledge Base of Agile Practices*, and the *Strategic Analysis Process* (Figure 1). The core of the framework is the Strategies Graph, inspired by the Strategy Map concept from Balanced Scorecards (BSC) in strategic management [6].The fundamental idea in BSC is to attain a balanced state in dealing with strategic objectives. Similarly, the SAAP framework highlights the importance of keeping balance among the various types of strategic goals in an organization while adopting a new software process. The SAAP framework was developed in response to strategic needs in one of the R&D units at Ericsson Software Research. In this paper, we introduce the SAAP framework with illustrations from the Ericsson experience.

2 The (SAAP) Framework

Figure 1 shows the main components of the framework. In the first phase of the *Strategic Analysis Process*, important strategic goals of the organization are extracted, classified, and visualized. Then, the strategic knowledge of candidate practices is retrieved from the pre-developed knowledge base of agile practices. The knowledge base contains knowledge collected from empirical studies about how each agile practice contributes to different strategic goals under various project conditions. The developed Strategic Graph is used along the second phase of the strategic analysis process, in order to situationally analyze the strategic impacts of every candidate agile practices; as well as their overall impact as a new agile process.

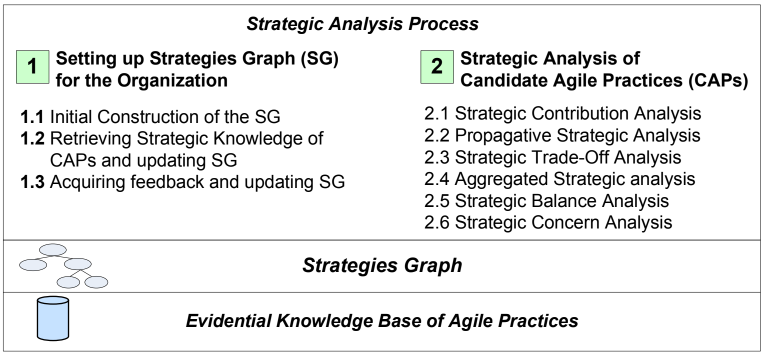

Strategic Analysis Process
1 Setting up Strategies Graph (SG) for the Organization
1.1 Initial Construction of the SG
1.2 Retrieving Strategic Knowledge of CAPs and updating SG
1.3 Acquiring feedback and updating SG
2 Strategic Analysis of Candidate Agile Practices (CAPs)
2.1 Strategic Contribution Analysis
2.2 Propagative Strategic Analysis
2.3 Strategic Trade-Off Analysis
2.4 Aggregated Strategic analysis
2.5 Strategic Balance Analysis
2.6 Strategic Concern Analysis
Strategies Graph
Evidential Knowledge Base of Agile Practices

Figure 1: Overview of SAAP Framework

2.1 *Phase 1:* Setting up the Strategies Graph for the Organization

The Strategies Graph (SG) expresses the *decompositional* and *contributional* relations of strategies at different levels of organization. Decompositional relations represent the AND/OR decomposition of high-level strategies to low-level objectives. The contributional relations represent the kind of impacts that strategic objectives might have on each other. The upper part of Figure 2 shows a portion of the SG, developed in one of the experiments of SAAP.

The Strategies Graph adopts its main constructs from the *i** modeling framework [7]. *i** is a goal and agent oriented modeling framework which can be used to represent the strategic aspects of a modeling domain. The *i** concept of Softgoal is used to model strategic objectives. The contributional relations of strategic objectives are represented by a variant of *i** notation of Contribution Link: "++" For Strong Positive, "+" for Positive, "-" for Negative, and "--" for Strong Negative contributions. "AND" and "OR" links are used to represent logical decomposition of strategic objectives.

[Step 1.1] Initial Construction of the Strategies Graph

The first step in applying SAAP is to develop the SG. The initial version of SG is developed by selected members of the Analysis Team. The framework stresses the participation of representatives all organizational roles. A participatory approach is needed to bring various stakeholders' viewpoints into a model of the organization's strategies. The role of middle management representatives is crucial for creating the SG. The initial version of SG often contains the strategic objectives that matter most to the organization, and which are not well supported by the as-is development process.

[Step 1.2] Retrieving Strategic Knowledge of CAPs and Updating SG

The second step of SAAP is to enrich the Strategies Graph of organization with the strategic objectives, which are tightly bound to agile values. The SAAP framework is built on top of an evidential knowledge base of agile practices. This knowledge base (which was introduced in an earlier paper [5]) contains the strategic information of agile practices. The contents of this knowledge base have been collected by systematic review of extensive number of empirical studies, which had reported the behavior of different agile practices in various project situations. Therefore, the strategic objectives that are presented for each agile practice are all supported by references to peer-reviewed empirical research papers. Indeed, the content of this knowledge base is evidence-based as it provides a brief description of the situation in which a particular contribution from a practice to an objective was observed. This knowledge base is available online at www.ProcessExperience.org.

The SAAP framework uses the content of the content of the knowledge base for completing the strategies graph of organizations. The reason for incorporating the built-in strategic objectives of agile practices into the strategic model of the organization is rooted to the intention of organization for adopting agile. Such organizations should have a clear understanding of agile objectives, and find a right place of those objectives within their organizational strategic model. For instance, in our experiment, one of the strategic objectives of the R&D unit (which was expected to be improved) was the "Reduced Development Cost" (shown in Figure 2). The knowledge base of agile practices introduced a number of related objectives, defined in the Lean method, which by focusing on "Avoiding Waste" positively contributes to the “Reduced Development Cost” objective. The content of this knowledge base will be also used in the later steps of the framework.

[Step 1.3] Acquiring Feedback and Updating the SG
The Strategies Graph is developed iteratively. In our experience at Ericsson, the initial version of SG was developed by selected members of the analysis team, and updated with the strategic knowledge of agile practices. Afterwards, the SG is passed to other members of the analysis team, as well as other organizational members in order to get feedbacks and complete the model. Group meeting is indeed an effective approach for completing the SG, by reflecting opinions of different organizational parties.

2.2 *Phase 2:* Strategic Analysis of Candidate Agile Practices

The purpose of this phase is to investigate impacts of candidate agile practices on the strategic objectives of organization. This framework takes a model-driven approach for the strategic analysis of candidate agile practices, and uses the Strategies Graph of the organization as the basis of most analyses activities. The framework introduces five types of strategic analysis:

[Step 2.1] Strategic Contribution Analysis
The foremost step of strategic analysis is to explore *contributions* of every Candidate Agile Practice (CAP) towards the organizational strategic objectives visualized on the Strategies Graph. As shown in the Figure 2, every contribution relation has two elements:

1. **Contribution Type** – For specifying how the CAP affects an objective. The framework, inspired by the *i** modeling framework, defines four types of contributions: *Strongly Positive* (++), *Positive* (+), *Negative* (-), and *Strongly Negative* (--), where in positive contributions the enactment of CAP would help the achievement of objective, and vice versa for negative ones.
2. **Contribution Rationale** – For specifying why the CAP affects the objective. For example, when a CAP like “Scrum Team Structure” is identified to be making Positive (+) contribution to the objective “Avoid Extra Features”, its rationale is that “sell-organizing members of a Scrum team can better identify extra features and decide on their removal or replacement”.

Two approaches are proposed for deriving the contribution relations: *evidence-based* or *consensus-based*. It is evidence-based if the strategic objective appears among the retrieved strategic knowledge of the CAP. Thus, the type and rationale of contribution can be extracted from the knowledge base. When the evidence is unavailable, or is judged to be inadequate or unreliable, the analysis team would take a consensus-based approach to derive this contribution relation, based on the original definition of the CAP.

In specifying the type of a contribution relation, the analysis team should consider the possibility of *situational behaviors*. It is possible that a CAP, in some particular situations, impacts an objective differently from its general behavior. For example, the contribution of the CAP “Pair Programming” towards the objective “Be On-time to Market" is situational, in that in some cases the CAP would help, and in some other cases in would hurt the objective. This information is retrieved from the Knowledge Base of SAAP. In this example, the knowledge base states that "when the market pressure is not high, and there is adequate number of developers, pairing programmers would help the project to be on time for market, whereas in other cases it hurts." Knowing the situational behaviors of a CAP towards an objective allows the analysis team to choose contribution values that are best matched with their own organization and project context.

[Step 2.2] Propagative Strategic Analysis

Propagative Strategic Analysis allows anticipating the impacts of an agile practice on higher-level strategic objectives. To perform this analysis, the value of contribution relations will be propagated along the strategies graph. For instance, as shown in Figure 2, enacting the CAP "Scrum Team Structure" would make positive contribution to the objective "Reduced Waiting Time", which consequently makes positive impacts over strategic objectives: "Avoid Waste", and "Reduced Development Cost". The propagative analysis of SAAP is based on the *i** forward propagation algorithm [8].

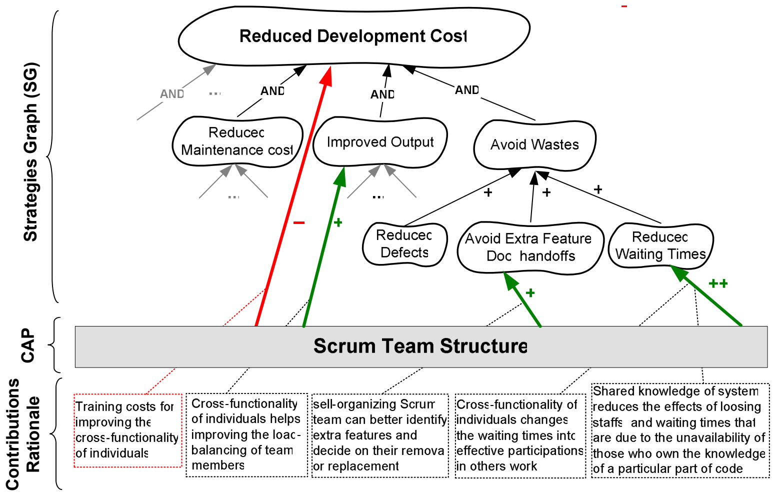

Strategies Graph (SG)
CAP
Contributions Rationale
-
Reduced Development Cost
AND
...
AND
AND
AND
Reducec Maintenance cost
Improved Output
Avoid Wastes
...
...
-
+
+
+
+
Reducec Defects
Avoid Extra Feature Doc handoffs
Reducec Waiting Times
+
++
Scrum Team Structure
Training costs for improving the cross-functionality of individuals
Cross-functionality of individuals helps improving the loac-balancing of team members
sell-organizing Scrum team can better identify extra features and decide on their remova or replacement
Cross-functionality of individuals changes the waiting times intc effective participations in others work
Shared knowledge of system reduces the effects of loosing staffs and waiting times that are due to the unavailability of those who own the knowledge of a particular part of code

Figure 2: Strategic Contribution Analysis of the Candidate Agile Practice (CAP) "Scrum Team Structure" to a portion of the SG of our experiment case

[Step 2.3] Strategic Trade-Off Analysis

Strategic Trade-Off Analysis allows comparing alternative agile practices with respect to their contributions to the strategic objectives of an organization. In SAAP, alternative practices are compared with respect to their positive and negative contributions to the strategic goals of organizations, and the significance of every contributed goal. For instance, "Pair Programming" and "Peer Review" are two alternative practices that are often suggested for "Reducing Defect Rate" in source code. However, there are other strategic goals which will be influenced by the enactment of any of these practices in an organization, depending to the project situation, e.g., “Cost of Development”, “Time to Market”, “Productivity of Individuals”, “Novice Developers’ Training”, and “Knowledge Sharing”.

SAAP uses a model-driven approach for trade-off analysis, and benefits from the *Propagative Strategic Analysis*. In this regard, the trade-off analysis would be performed not only with respective to the lower-level objectives, but also for the higher-level strategies of the organization. One approach for trade-off analysis in goal graphs is presented in [9].

[Step 2.4] Aggregated Strategic analysis

The purpose of aggregated strategic analysis is to explore the overall impact of the new agile method over the strategic objectives of an organization. In this analysis, for each organizational strategic objective, all the contributions from all candidate practices of new method are combined to produce the contribution of new agile method to that specific objective. After aggregation of contribution relations, every organizational strategic objective will take one of the following statuses:

- *Supported* – received homogeneous positive contributions
- *Declined* – received homogeneous negative contributions
- *Strongly Supported* – a supported objective with strongly positive contributions
- *Strongly Declined* – a declined objective with strongly negative contributions
- *Conflicted* – received heterogeneous contribution types from different practices
- *Unaddressed* – not contributed to by any practice, neither directly nor indirectly

[Step 2.5] Strategic Balance Analysis

Following Balanced Scorecards, one of the goals of the SAAP framework is to investigate whether the new agile method makes a balanced contribution to all categories of objectives. More specifically, in this framework, the transition to a new method is considered to be unbalanced if its positive contributions to one category of strategic objectives lead to significant bad effects on some other category of objectives. The balance of a transition does not imply that the selected set of practices is the optimum set, but an optimum set should make balanced impact over the strategic objectives. In [10] we introduced the concept of *Strategically Balanced Process Adoption* (SBPA), and specified its details. The SBPA considers a process adoption to be balanced, provided that it meets the following conditions:

1. It positively contributes to the strategic objectives, which are expected to be improved.
2. It does not cause uncontrolled negative impacts on the strategic objectives, which are not within the focus of improvement.
3. It does not cause overall deterioration of a particular category of strategic objectives, for the sake of improving some other categories.
4. It results in homogenous impacts over all categories of strategic objectives.

Detailed algorithms have been proposed in [10] to anticipate the attainability SBPA criteria.

[Step 2.6] Strategic Concern Analysis

Software process improvements are often motivate by the emergence of inefficiency symptoms in the current development process. These symptoms in a broader sense can be referenced in terms of *as-is process concerns*. When designing a new (to-be) development process, organizations should have an understanding of whether it will properly address their current concerns. SAAP is proposing the *Strategic Concern Analysis* in order to first, investigate the impacts of as-is process concerns on the strategic objectives of organization, and second, analyze whether the candidate set of agile practice would address the existing process concerns. The result of this analysis is key to the acceptance of CAPs, as if they fail to address the current concerns they cannot form an effective process.

To investigate the impacts of current process concerns on the strategic objectives of the organization, a similar approach of [step 2.1] can be applied. In this approach the identified process concerns are visualized next to the SG, and their negative contributions to the strategic objectives are investigated. This activity also requires the participation of representatives of different organizational roles, in order to come up with a right set of strategic objectives, which are affected by every process concern. The model driven approach (the visual aid of SG) facilitates this activity, and reduces the overhead of analysis.

To analyze whether the current set of CAPs are addressing as-is process concerns, the strategic contribution models of CAPs and process concerns is used. This analysis is based on the heuristic that when a strategic objectives is negatively contributed by a process concern PC_i, and positively contributed by the candidate agile practice CAP_j, it is possible that the CAP_j strategically addresses the PC_i. Further analyses of CAPs in regard with the as-is process concerns, requires root-cause analysis of process concerns, and investigation of the impacts of every CAP on the roots of process concerns.

3 Discussion and Future Work

The importance of acting strategically in transition to agile would become apparent when we observe the change of a method as a consequential strategic decision, which influences not only the technological, but also business and organizational objectives of an organization. The proposed framework of Strategic Analysis of Agile Practices (SAAP) investigates the impacts of a new agile method on organizational strategic objectives. The SAAP framework is proposed for the early stages of transitioning to agile, where organization would decide on the trade-offs of new method. The approach of this framework in the strategic analysis of agile practices is inspired by the idea of Balanced Scorecards [6], which emphasizes the establishment of organizational strategic model as the basis of a decision making framework in an organization.

The SAAP framework can be combined with most of the current frameworks of transition to agile, and complement their lack of attention to the strategic aspects of the transition process. It can be also used as a stand-alone framework for strategic analysis of a set of candidate agile practices, in order to find their potential compliance and conflicts with strategic interests of an organization.

A number of issues have been identified as threats to the validity of the results of SAAP framework, which some of them can be mitigated. The reliance of framework to the knowledge base on agile practices can pose a risk to the framework, as there might not adequate information about all of the agile practices. However, this knowledge base in under expansion, and will cover a wider range of agile practices in future. The other risk to the SAAP is *Over-Pessimistic or -Optimistic Evaluations* – where there is no evidence for the contribution of an agile practice to a strategic objective, yet the contribution is perceived possible, in some cases the subjective evaluations might be unrealistic. Of course the level of familiarity and experience of chief members of Analysis Team in regards with agile practices and their built in objectives can influence the validity of Analysis results.

As for future work, the framework is going to be expanded for covering the full lifecycle of transitioning to agile. The framework has been tested so far in one study, further case studies will be an essential part of future work.

References

1. A. Qumer and B. Henderson-Sellers, A framework to support the evaluation, adoption and improvement of agile methods in practice.In Journal of Systems and Software, 81(11) p. 1899-1919 (2008)
2. A. Sidky, J. Arthur, and S. Bohner, A disciplined approach to adopting agile practices: the agile adoption framework.In Innovations in Systems and Software Engineering, 3(3) p. 203-216 (2007)
3. I. Krasteva, S. Ilieva, and A. Dimov, Experience-Based Approach for Adoption of Agile Practices in Software Development Projects, in Advanced Information Systems Engineering, Springer Berlin / Heidelberg. p. 266-2801 (2010)
4. J. Ralyté, R. Deneckère, and C. Rolland, Towards a Generic Model for Situational Method Engineering, in Advanced Information Systems Engineering. p. 1029-10291 (2003)
5. E.H. Chiniforooshan, E. Yu, and M.C. Annosi. Itemized Strategic Dependency: a Variant of the i* SD Model to Facilitate Knowledge Elicitation. In 4th International i* Workshop. Tunis (2010)
6. R.S. Kaplan and D.P. Norton, The balanced scorecard: Translating strategy into action, Boston, Harvard Business School Press (1996)
7. E.S.K. Yu. Towards modelling and reasoning support for early-phase requirements engineering. In Proceedings of the Third IEEE International Symposium on Requirements Engineering: IEEE Computer Society (1997)
8. J. Horkoff and E. Yu. Using the i* Evaluation Procedure for Model Analysis and Quality Improvement presentation. In Second International Workshop on i* / Tropos. University College London, London UK (2005)
9. G. Elahi and E. Yu, A Goal Oriented Approach for Modeling and Analyzing Security Trade-Offs, in Conceptual Modeling - ER 2007. p. 375-3901 (2008)
10. H. Chiniforooshan Esfahani, E. Yu, and M.C. Annosi. Strategically Balanced Process Adoption. In International Conference on Software and System Processes (ICSSP'11). Hawaii, USA: ACM (2011)

Improving Agility in Model-Driven Web Engineering

José Matías Rivero[1,2], Julián Grigera[1], Gustavo Rossi[1,2], Esteban Robles Luna[1], Nora Koch[3,4]

[1] LIFIA, Facultad de Informática, UNLP, La Plata, Argentina
{mrivero, julian.grigera, gustavo, esteban.robles}@lifia.info.unlp.edu.ar
[2] Also at Conicet
[3] Ludwig-Maximilians-Universität München, [4] Cirquent GmbH, Germany
kochn@pst.ifi.lmu.de

Abstract. The increasing growth of the Web field has promoted the development of a plethora of Model-Driven Web Engineering (MDWE) approaches. These methodologies share a top-down approach: they start by modeling application content, then they define a navigational schema, and finally refine the latter to obtain presentation and rich behavior specifications. Such approach makes it difficult to acquire quick feedback from customers. Conversely, agile methods follow a non-structured, implementation-centered process building software prototypes to get immediate feedback. In this work we propose an agile approach to MDWE methodologies (called Mockup-Driven Development, or MockupDD) by inverting the development process: we start from user interface mockups that facilitate the generation of software prototypes and models, then we enrich them and apply heuristics in order to obtain software specifications at different abstraction levels. As a result, we get an agile prototype-based iterative process, with advantages of a MDWE one.

Keywords: Mockups, User-Interface, Agile, Web Engineering, MDD

1 Introduction

During the last 20 years, many Model-Driven Web Engineering (MDWE) methodologies have been defined to improve the development process of web applications approaches [1-4]. All of these methodologies share a common top-down approach [5] and construct web applications by describing a set of models at different abstraction levels:

- *Content (or Domain) Model*: defining domain objects and their relationships.
- *Hypertext (or Navigation) Model*: defining navigation nodes and links that publish information specified by objects in the *Content Model*.
- *Presentation Model*: refining the *Hypertext Model* with concrete user-interface presentation features like pages, concrete widgets, layout, etc.

This process is generally top-down, delivering a final web application through a process of (sometimes automatic) model transformations which maps the previously described models into other models or a specific technology.

Agile methodologies, on the other hand, promote early and constant interaction with customers to assert that the software built complies with their requirements, by constantly delivering prototypes developed in short periods of time. Agile approaches argue that software specifications must emerge naturally, enhancing former prototypes along the development until the final application is obtained.

To summarize, while MDWE methodologies facilitate software specification portability, abstraction and productivity, they fail in providing *agile* interaction with customers because concrete results are obtained too late. On the other hand, while this feature is clearly provided by agile methodologies, they are heavily based on direct implementation and thus fail to provide abstraction, portability and productivity through automatic code-generation.

In this paper we propose an hybrid model-based agile methodology – called Mockup-Driven Development (MockupDD) – aiming to extract the best of both worlds, i.e. a process driven by the active participation of users and customers, and a classical approach following the phases of analysis, design and implementation assisted with the use of models in all stages. Our approach starts by the requirement analysis, i.e. defining mockups (ideally together with the customers) to agree upon the application's functionality, similar to Harel's behavioral programming approach [6]. Then, mockups are translated to an abstract user-interface model that can be directly derived to specific MDWE presentation models or technology-dependent UI prototypes. By tagging mockups and presentation models we add navigation features, and based on the navigation specification, we use heuristics to infer content models. Thus, we are starting the requirement specifications with objects that are perceivable by customers (UI structure elements), easing requirements gathering and traceability [7].

Therefore, since we start with presentation models obtained from mockups and then construct or obtain *upper* (i.e. abstract) models, we are inverting the traditional MDWE process, yielding to a more *agile*, yet truly model-based approach. While we exemplify with the UML-based Web Engineering (UWE) [3], MockupDD can be applied to any MDWE approach.

2 MockupDD by Example

User Interface (UI) Mockup tools like Balsamiq, Pencil or Mockingbird[1] suit well in agile methodologies [8-10], since they provide a quick and easy way of capturing interaction requirements. Usually, mockups are defined in companion with other specifications like use cases [11, 12], user stories [13] or informal annotations [14]. Also, mockups have been introduced in the context of model-driven development (MDD) approaches like ConcurTaskTrees [15]. In most cases, however, mockups themselves are not considered as models and they are usually thrown away after requirement modeling. Thus, mockups are not used as important drivers of the development process although they contain precise information about the users' needs.

MockupDD starts the development process by creating UI mockups with a mockup tool. As we have shown in a previous work [16], the resulting mockup files can be

[1] http://balsamiq.com, http://pencil.evolus.vn/en-US/Home.aspx, https://gomockingbird.com, last visited 18.3.2011

parsed and translated to an abstract UI model called *SUI model* (Structural UI Model) that can be in turn translated to presentation models of modern MDWE methodologies through a simple mapping, since most presentation metamodels (SUI included) usually share the same concepts (e.g., pages, panels, links, buttons, etc.). We propose to enrich SUI models using *tags*. Tags define simple but precise specifications that are applied over particular types of SUI elements and represent hints that can result in the derivation of particular MDWE model concepts.

In this paper we introduce *navigation tags* that enrich SUI models in order to derive navigation models. After obtaining both presentation and navigation models by the aforementioned mapping and tags semantics respectively, we apply heuristics to obtain the content model as well. We illustrate our process by showing how it works in the context of the development of a music catalogue application, deriving models for the UWE methodology. We have chosen UWE because it is representative of an important group of methods, it is based on UML and it has tool support. A schematic diagram of our process is shown in Figure 1.

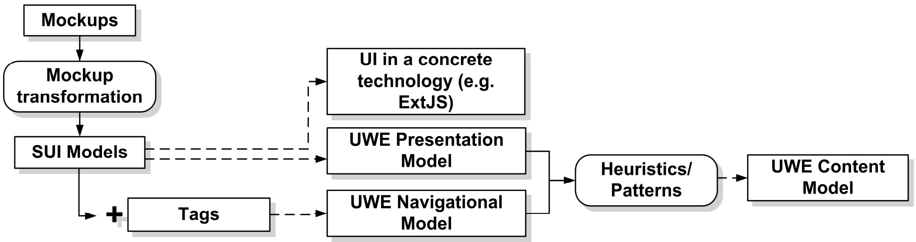
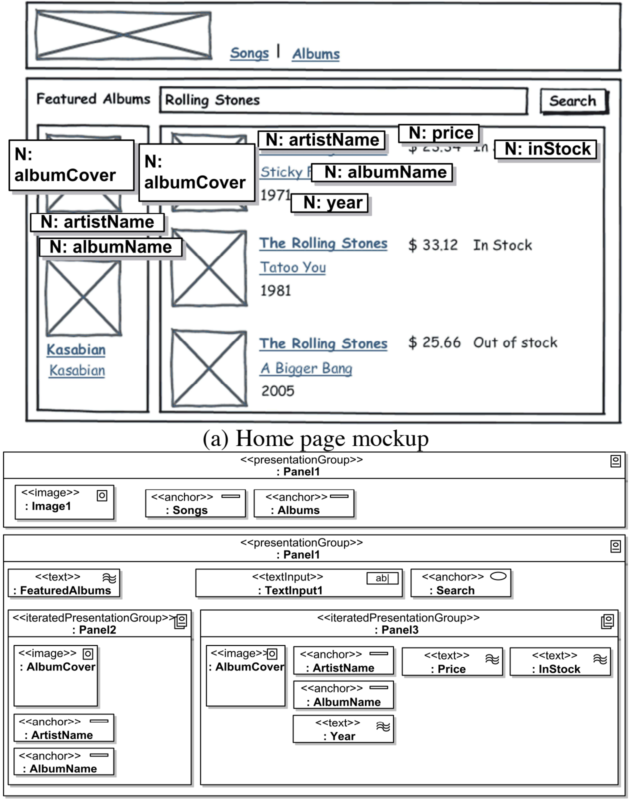


Figure 1. Mockup-Driven Development (MockupDD) process.

2.1 From Mockups to Presentation Models

The UI mockup (shown in Figure 2.a) depicts the home page of the Music Portal application containing a header, a list of featured albums, an album search box and its corresponding search result. Figure 2.b shows the corresponding UWE presentation model that can be obtained through a simple SUI-to-UWE presentation widget mapping. Some advanced features (like choosing whether to use an UWE `PresentationGroup` or an `IteratedPresentationGroup`) are inferred during the mockup transformation process through mockup analysis. The first problem that emerges is that the name of some widgets cannot be inferred; in these cases, a generic id is generated (like `Panel1`, `TextInput1` or `Image1`). Since correctly naming model elements with identifiers is important to reference them in the future and also for code or model derivation, we define a *naming tag set*, that allows redefining the name of some widgets when needed. The tagged mockup and resulting UWE presentation model are shown in Figure 2; note that naming tag starts with an `N:`. The use of naming tags implies that correct names are stored associated with SUI model elements and thus reflected in derived MDWE presentation ones. Also, when correctly applied, naming tags allow deriving mockup implementations for concrete technologies like ExtJS[2] using natural widget ids as when working directly with code.



(a) Home page mockup

(b) Generated UWE presentation model after applying naming tags

Figure 2. Deriving an UWE presentation model from a mockup.

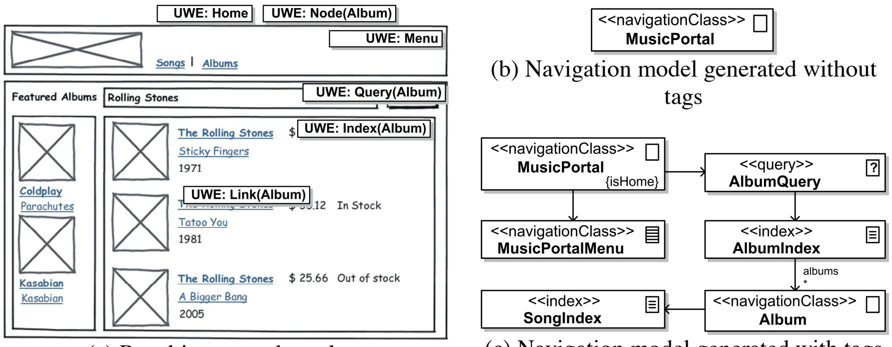
2.2 Deriving Navigational Models

After deriving presentation models, a naive approach to start generating navigation models could be defining one UWE `NavigationClass` (the UWE navigation concept for defining nodes) for each mockup. However, the UWE metamodel defines several navigation elements in addition to elements of type `NavigationClass`: `Query`, `Index` and `Menu`. While `Queryes` and `Indexes` represent information retrieval and selection of a particular element in a collection respectively, `Menus` are used to specify alternative navigation paths.

Since we cannot directly infer which UWE navigation element must be used in every mockup (this election requires design or modeling skills), we have defined a second tag set: the UWE navigation tag set. This set contains a tag for every UWE

[2] http://www.sencha.com/products/extjs/, last visited 18.3.2011

navigation element. Figure 3 shows the resulting tagged mockup and the consequences of tag application in derived UWE navigation model.



(a) Resulting tagged mockup

(b) Navigation model generated without tags

(c) Navigation model generated with tags

Figure 3. Initial mockup with UWE navigation tags applied and the resulting navigation model.

The UWE navigation tags introduced are the following:

- `Home`: defines that the `NavigationClass` related to the mockup is the home of the navigation model.
- `Node(<nodeId>)`: Assigns an id to the `NavigationClass` related to the mockup in order to be referenced as the destination of one or more navigation (`Link`) tags.
- `Link(<nodeId>)`: Specifies a navigation link to another `NavigationClass`. A corresponding `Node` tag with the same `<nodeId>` must be specified in order correctly derive the navigation.
- `Query(<elementId>)` and `Index(<elementId>)` define a `Query` involving elements of type `<elementId>` and the `Index` in which the results of the `Query` are shown.
- `Menu` specifies that the panel over which it is applied is a set of links, a so called UWE `Menu`.

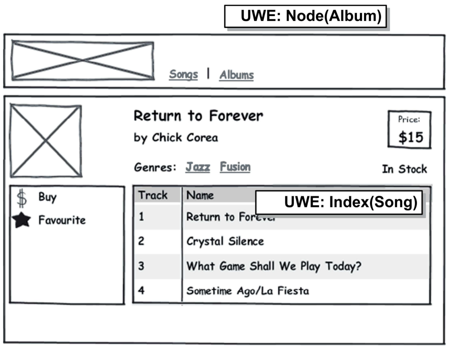


Figure 4. Album details mockup with UWE navigation tags applied.

When clicking on an album's title in the home page, an UI of the album details will be shown. A mockup of such user interface is denoted in Figure 4. The complete UWE navigation model can be observed in the already introduced Figure 3.c in which the `Album NavigationClass` is included. The navigation link is expressed through the `Link(Album)` and `Node(Album)` tags in home page and album mockups, respectively.

2.3 Towards a Content Model

Once we have obtained the UWE navigation model, a first version of the content model can be derived by applying some inference rules described in Figure 5. These rules were designed by studying many examples of UWE navigation and content models and discovering recurrent patterns in them.

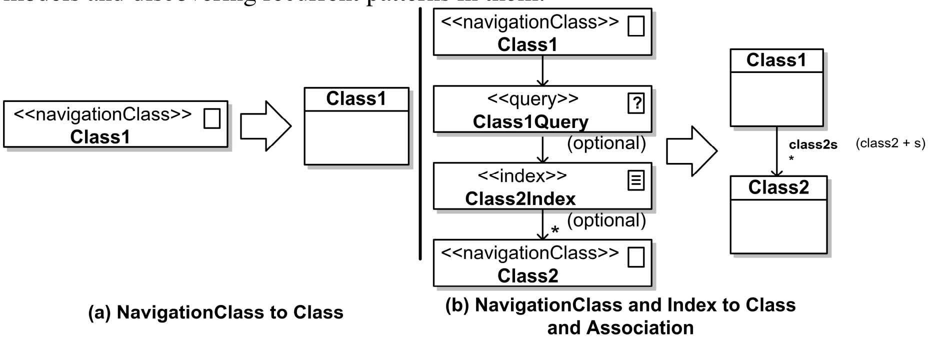


Figure 5. Two content inference rules.

UWE navigation element names (previously generated using naming and UWE navigation tags) are used to derive the names of the content elements. The resulting UWE content model after the application of the introduced rules over the UWE navigation model of Figure 3.c is shown in Figure 6 (for space reasons, only a part of the navigation model is shown).

The obtained UWE content models must be refined in order to specify class attributes. As UWE navigation models do not allow more refinement than the features already commented, this information should be taken from other models. Since in UWE every navigation concept is refined by a presentation specification (e.g., a `PresentationGroup`), and given that we have already derived these models from SUI specifications, we can use this link between models in order to obtain attributes from presentation structure. An example of this approach is denoted in Figure 7.

Automatic derivation may naturally lead to an imprecise content model, and some thoughtful design might be required from a developer in order to get to a definitive version. However, even when most design adjustments can not be fully automated, they can be still predicted. For example, an album presentation model might translate into an album class with attributes such as *artistName*, when in fact the content model should have two separate classes for `Album` and `Artist`, related to each other. We have observed that many of these inaccurate derivations usually repeat, so the required adjustments can be documented (and applied with automatic assistance when possible) just like code refactorings [17].

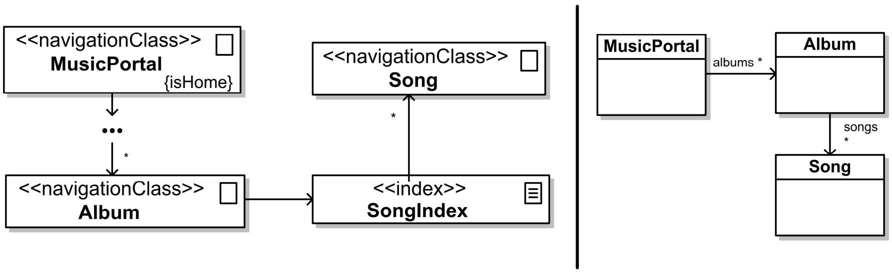


Figure 6. Inferred UWE content model derived through the application of the introduced rules.

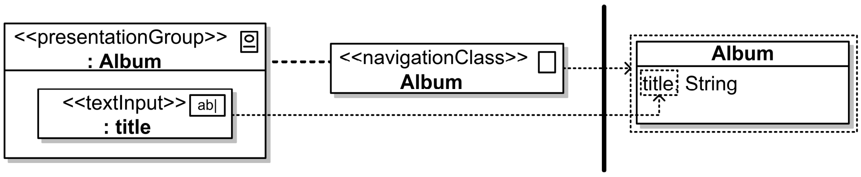


Figure 7. Attribute inference from presentation specifications.

3 Conclusion and Further Work

We have presented a mockup-based approach (MockupDD) pursuing an inversion of the traditional MDWE process. We decided to start our process with mockups because they are becoming a common tool in agile methodologies to interact and establish a shared view of requirements between customers and developers. Mockups are processed to structured UI models (called SUI) and with the help of tags they are easily derived to MDWE presentation and navigation models. Applying a set of inference rules, a first version of MDWE content models can be generated. We have shown the approach applied to a brief example using the UWE methodology. With our approach, we intend to provide an agile methodology based on UI mockups and lightweight specifications to obtain MDWE models, which offer advantages like automatic code generation.

Extending the proposed approach to other modern MDWE methodologies like WebML represents a fruitful work path. We are interested in defining a general and methodology-agnostic navigation tag set that also allow deriving navigation models for a more comprehensive set of MDWE approaches. Finally, since obtained content models likely require to be refactorized, we are interested in developing heuristics to suggest refactoring alternatives to be applied over content specifications.

4. References

1. Ceri, S., Fraternali, P., Bongio, A.: Web Modeling Language (WebML): A Modeling Language for Designing Web Sites. Computer Networks and

ISDN Systems, 33(1-6), pp. 137-157 (2000)

2. Gómez, J. and Cachero, C.: OO-H Method: Extending UML to Model Web Interfaces (2003). In: Information Modeling For internet Applications, pp. 144-173, P. van Bommel, Ed. IGI Publishing, Hershey, PA (2003)
3. Koch, N., Knapp, A.. Zhang G., Baumeister, H.: UML-Based Web Engineering, An Approach Based On Standards. In: Web Engineering, Modelling and Implementing Web Applications, pp. 157-191. Springer (2008)
4. Rossi, G., Schwabe, D.: Modeling and Implementing Web Applications using OOHDM. In: Web Engineering, Modelling and Implementing Web Applications, Springer, pp. 109-155 (2008)
5. Wimmer M., Schauerhuber, A., Schwinger, W., Kargl, H.: On the Integration of Web Modeling Languages: Preliminary Results and Future Challenges. In: Proc. of the 3nd Int. Workshop on Model-Driven Web Engineering (MDWE'07), CEUR-WS (2007)
6. Harel, D.: Some Thoughts on Behavioral Programming. In: Applications and Theory of Petri Nets. Springer Berlin Heidelberg (2010)
7. Seyff, N., Graf, F., Maiden, N.: End-user requirements blogging with iRequire. In: 32nd ACM/IEEE International Conference on Software Engineering - ICSE '10. ACM Press, New York, New York, USA (2010)
8. Noble J., Biddle, R., & Martin, A.: The XP Customer Role in Practice: Three Studies. In: Agile Development Conference, pp. 42-54. IEEE Computer Society (2004)
9. Ferreira J., Noble J., & Biddle R.: Agile Development Iterations and UI Design. In: AGILE 2007 Conference, Washington, DC: IEEE Computer Society, pp. 50-58 (2007)
10. Ton, H.: A Strategy for Balancing Business Value and Story Size. In: Agile 2007 Conference. Washington, DC: IEEE Computer Society, pp. 279-284 (2007)
11. Kulak, D. & Guiney, E.: Use Cases: Requirements in Context. Addison-Wesley (2004)
12. Homrighausen, A., Six, H., & Winter, M.: Round-Trip Prototyping Based on Integrated Functional and User Interface Requirements Specifications. In: Requirements Engineering, 7(1), pp. 34-45 (2002)
13. Cohn, M.: User Stories Applied: for Agile Software Development. Addison-Wesley (2004)
14. Moore, J. M.: Communicating Requirements Using End-User GUI Constructions with Argumentation. In: 18th IEEE International Conference on Automated Software Engineering, pp. 360-363, IEEE Computer Society (2003)
15. Panach, J. I., España, S., Pederiva, I., & Pastor, O.: Capturing Interaction Requirements in a Model Transformation Technology Based on MDA. Journal of Universal Computer Science, 14(9), pp. 1480-1495 (2008)
16. Rivero, J. M., Rossi, G., Grigera, J., Burella, J., Robles Luna, E., Gordillo, S. E.: From Mockups to User Interface Models: An Extensible Model Driven Approach. In: 10th International Conference on Web Engineering, pp. 13-24. Springer (2010)
17. Fowler, M., Beck, K., Brant, J., Opdyke, W., Roberts, D.: Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional (1999)

Comprehension and Utilization of Core Assets Models in Software Product Line Engineering

Iris Reinhartz-Berger[1], Arnon Sturm[2], and Arava Tsoury[1]

[1] Department of Information Systems, University of Haifa, Haifa 31905, Israel
iris@is.haifa.ac.il, aravabt@gmail.com
[2] Department of Information Systems Engineering,
Ben-Gurion University of the Negev, Beer Sheva 84105, Israel
sturm@bgu.ac.il

Abstract. In software product line engineering, *core assets* are reusable artifacts that are intended to be used by a family of software products in order to improve development productivity and quality of particular software products. In order to support the construction and maintenance of core assets, various modeling methods have been proposed. However, the assessment of these methods is still in an incubation stage. In fact, only several frameworks for comparing and evaluating these methods have been suggested. These mainly refer to lists of criteria whose examination is sometimes subjective and opinion-dependent. In this paper, we call for empirical evaluation of the comprehension and utilization of core assets and report the initial results of a series of studies we performed in this context.

Keywords: variability management, software product line engineering, domain analysis, UML, feature-orientation, evaluation

1 Introduction

In Software Product Line Engineering (SPLE), a *core asset* is "a reusable artifact or resource that is used in the production of more than one product" [5]. The development of core assets intends to improve productivity, increase quality of individual products, decrease development cost, decrease time to market or to launch new products or versions, and enable moving into new markets in shorter times. Core assets have different forms that may be useful in a software production process, one of which is domain models. These models capture both existing commonality and allowed variability of given product lines.

Reviewing 97 papers that describe variability management approaches in SPLE, reported from 1990s to 2007, Chen and Babar [4] conclude that the main corpus of approaches focuses on variability modeling and utilizes feature models (33 works) or UML and its extensions (25 works) for this purpose. Feature-oriented methods, such as [6], [13], [14], and [21], support specifying domain models as sets of characteristics relevant to some stakeholders and the relationships and dependencies among them. Variability is specified in terms of mandatory vs. optional features, alternatives, OR features, 'require' and 'exclude' dependencies among features, feature groups, and composition rules. UML-based SPLE methods (e.g., [10], [18], [20], [25], and [26]) usually suggest profiles for handling variability-related issues, including specification of mandatory and optional elements, dependencies among elements, variation points, and possible variants. Some UML-based methods suggest extending UML or representing variability aspects orthogonally to "regular" UML models of the product families, e.g., [11].

As the number of suggested modeling methods increases, several evaluation frameworks have been proposed for comparing methods, belonging to different categories, e.g., [9], [12], [15], and [24], or within certain categories, e.g., [8]. Matinlassi [15], for example, refers to four main comparison criteria: context, user, contents, and validation. Haugen et al. [12] suggest examining the ways variability and commonality are modeled, the support for iterative and incremental system family development, and the production of individual systems. Djebbi and Salinesi [8] compare feature-oriented notations in terms of different criteria, including readability, simplicity and expressiveness, adaptability, scalability, and others. Although the various criteria may help understand the benefits and limitations of the different methods, their usage in examining and comparing the methods is limited as they are subjective and usually criticized as opinion-oriented [4].

Despite their amenability to be empirically evaluated, relatively minor attention is allocated for the empirical evaluation of SPLE methods in general and variability management approaches in particular. These studies highlight different aspects in SPLE, including product derivation [22], quality assurance [2, 7], and architecture process activities [1]. In this paper we draw a general evaluation framework for comparing core assets modeling methods. This framework, which refers to both specification and utilization aids, is used for better understanding the sources of difficulties of core assets modeling methods. In a series of three studies, we started examining the specification aids of modeling methods to clearly describe common and variable parts in core assets and the relevant utilization aids, which aim at guiding the developer in generating, deriving, and building valid software products. We report on some sources of difficulties we found in the comprehension and utilization of core assets models using feature-oriented and UML-based methods.

The remainder of this paper is organized as follows. Section 2 reviews the suggested dimensions for evaluation, whereas Section 3 describes two core assets modeling methods on which we conducted the evaluation so far and justifies their selection. Section 4 elaborates on the empirical studies and reports our initial results. Finally, Section 5 concludes and refers to future research directions.

2 Dimensions for Evaluating Core Assets Modeling Methods

When evaluating core assets modeling methods, two important dimensions can be identified: *specification*, which refers to the collection of aids required for specifying both existing commonality and allowed variability in a software product line, and *utilization*, which refers to the different means to use core assets in order to create particular software product in the domain.

The specification aids are further divided into commonality- and variability-related ones. *Commonality-related aids* are used for specifying aspects that all (or most of) the products in the line exhibit, while *variability-related aids* enable specifying added values that not all the products in the family include in the same way.

The utilization dimension refers to the aids needed in core assets modeling methods in order to improve the effectiveness and efficiency of creating particular product artifacts in certain domains. This includes guidance and validation. *Guidance* refers to the ways in which core assets can be used for specific needs (i.e., in the development or production of particular software products), while *validation* refers to the mechanisms and tools that may be provided by the modeling methods for enabling alignment of specific software products with the domain constraints and rules as specified in core assets.

Table 1 summarizes the main specification and utilization aids of feature-oriented and UML-based methods. In order to define sources of difficulties in specifying and modeling core assets in these categories, we used the suggested evaluation framework and conducted three studies (the focus of each study is depicted in Table 1). Due to the large number of methods in each category, we had to select specific methods for evaluation. Explanations on the selected modeling methods, as well as the reasons for their selection are provided next.

Table 1. Evaluation Framework for Core Assets Modeling Methods

Category	**Specification Aids**		**Utilization Aids**	
	Commonality	**Variability**	**Guidance**	**Validation**
Feature-oriented	Mandatory and optional elements, dependencies	Feature groups, alternatives, and OR-related features	Cardinality, rationale, constraints	Instantiation and configuration conformance
UML-based	Mandatory and optional elements, dependencies	Variation points, variants	Cardinality, openness, reuse mechanisms, binding time	Specialization and configuration conformance

Legend: Eval 1, Eval2, Eval 3

3 The Selected Modeling Methods: CBFM and ADOM

3.1 Feature-Oriented Methods and CBFM

In feature-orientation, ***features*** are defined as end-user characteristics of systems, or distinguishable characteristics of concepts that are relevant to some stakeholders of the concepts [13]. Features can be composed and decomposed into trees, where the edges represent dependencies between features. Some of the feature-oriented methods, such as [14], concentrate on commonality specification and do not explicitly specify variability. However, guidance is partially supported in these methods, mainly via XOR and OR constructs or via explicit textual constraints and guidelines. Some methods, e.g., [6] and [23], support representing variation points and variants via feature groups and refer to guidance via OR and XOR constructs.

Cardinality-Based Feature Modeling (CBFM) [6] exceeds the expressiveness of other feature-oriented methods by enabling usage of OCL for specifying different dependencies and allowing definition of various cardinalities for better guiding the development of particular software products. In particular, CBFM extends the expressiveness of feature diagrams in FODA [13], the ancestor of most feature-oriented methods, with five main aspects: (1) *cardinality*, which denotes how many clones of a feature can be included in a concrete product, (2) *feature groups*, which enable organizing features and defining how many group members can be selected at certain points, (3) *attribute types*, indicating that attribute values can be specified during configuration, (4) *feature model references*, which enable splitting a feature diagram into different diagrams, and (5) *OCL constraints*.

3.2 UML-based Modeling Methods and ADOM

Most UML-based methods model commonality-related aspects via dedicated stereotypes for differentiating mandatory (sometimes called kernel) and optional elements. Some works explicitly specify variability using both «variation point» and «variant» stereotypes, while others specify only one of these concepts and the other is implicitly specified from its relationships with the other concept.

We selected the Application-based DOmain Modeling (ADOM) method [18] for our evaluation, since it consistently integrate all the main stereotypes from other methods in the UML-based SPLE category [19] and it explicitly refers to guidance and validation of software products with respect to core assets, aspects which other methods in this category tend to neglect. Furthermore, it enables explicit specification of both variation points and variants and it allows specifying ranges of multiplicity.

At the basis of ADOM there is a profile that includes the following six stereotypes: (1) «multiplicity», specifying the range of product elements that can be classified as the same core element, (2) «variation point», indicating locations where variability may occur, including rules to realize the variability in these locations, (3) «variant», which refers to possible realizations of variability and is associated to the corresponding variation points, (4) «requires» and (5) «excludes», which determine dependencies between elements (and possibly between variation points and variants), and (6) «reuse», which is used for guiding the developer about the possible usages of the core asset element in specific products.

4 Empirical Evaluation of Core Assets Modeling Methods

4.1 Eval1: Comprehension and Utilization of ADOM's Models

The first study was associated with two research questions: (1) Are the specification aids of ADOM well understood and to what extent? (2) Are the specification aids of

ADOM well utilized and to what extent? The subjects of this study were 15 advanced undergraduate and graduate students in an Information Systems program at the University of Haifa, Israel, who took a seminar course named "Advanced Topics in Software Engineering" in 2009. During the course, the students studied domain engineering techniques, focusing on ADOM and its capabilities. The study took place towards the end of the course as a class assignment. The students got a domain model (in ADOM) and had to answer questions in three categories. In the first category of questions, which referred to *comprehension*, the subjects had to answer 14 true/false questions regarding the domain and explain their answers based on the given model. The questions referred to both commonality and variability aspects. The second group of questions, *validation*, required finding violations in a particular application, with respect to the domain constraints as specified in the given model. For checking this task, we prepared a list of 9 mistakes (or inaccuracies) in the application model and measured the performance of the subjects in terms of precision, recall, and F-measure. Finally, in the third part, *guidance*, the subjects were asked to model another application in the domain based on a list of requirements and the given domain model (in ADOM). In this part, we examined how the specification aids were utilized for guiding the creation of particular models.

The results of this study brought up the following main points. First, variant-related aspects are better comprehended than variation point-related aspects. Our conjecture regarding this observation is that variation points are more abstract, usually refer to several elements (variants) and include information regarding the way to realize the variability. Thus, their specification is more difficult to understand than that of variants, which are more concrete and focus on particular elements. Second, errors that referred to commonality-related aspects, including such that refer to optional elements and not just to mandatory ones, are easier to find than errors that referred to variability. Furthermore, variability-related errors that involved several different model elements were the most difficult to detect (only two students found one such error each). Third, the subjects had difficulties in mapping the particular application elements to the domain elements as specified in the domain model. As this mapping may reveal anchors for validation, these difficulties also prevent the subjects from correctly identifying problems that are related to both commonality and variability issues. Finally, we found a correlation between the success in applying a variation point and the success to utilize its variants. However, it seems that the guidance provided by variation points is less considered than the guidance provided by the variants. A possible reason for this may be again the different abstraction levels of variation points and variants.

4.2 Eval2: Specification and Guidance Aids in ADOM

The second study addressed two research questions: (1) Do variability specification and guidance aids help comprehend core assets and to what extent? (2) Do variability specification and guidance aids help create or model correct products and to what extent? The subjects of this study were 116 advanced undergraduate students in an Information Systems Engineering program at Ben-Gurion University of the Negev, Israel, who took a mandatory course named "Object-Oriented Analysis and Design" in 2009. During the course, the students studied the ADOM method and the study took place as part of the final exam in the course. The students were randomly divided into four groups, each of which got a core asset model (in ADOM) and had to answer 15 comprehension questions regarding the given model and to model a particular application in the domain. The model given to the first group included only commonality-related stereotypes, namely «multiplicity», «requires», and «excludes». The model given to the second group included guidance-related stereotypes (i.e., «reuse») besides the commonality-related stereotypes, while the model given to the third group included, besides the commonality-related stereotypes, variability-related stereotypes, namely «variation point» and «variant». The model given to the fourth group included all six stereotypes. Despite the difference in the sets of stereotypes provided to the four groups, the models included similar (equivalent) information using UML expressiveness and associating textual notes when required.

The following interesting points have risen from this study. First, the guidance aids, which explicitly explain how to reuse a core asset element in a particular software product, help comprehend aspects that refer to commonality and variability issues, and not just to reusability. This was especially remarkable when referring to variation points and their rules to select variants. Our conjecture is that explicit specification of guidance required additional attention from the students, thus resulting in better outcomes. Second, the existence of all stereotypes seemed to complicate the core asset models and negatively affect comprehension. Finally, no statistically significant differences were found among the particular models produced by the various groups from the core asset. We believe that this is due to the clear and unambiguous requirements of the requested system, a situation which is less realistic in "real life", but required for a controlled experiment.

4.3 Eval3: Comprehension of CBFM and ADOM Specification Aids

The research question in the third study was: The specifications of which method, out of CBFM and ADOM, are more comprehensible and to what extent? The subjects in this study were 18 advanced graduate and undergraduate information systems students at the University of Haifa, Israel who took the seminar course “Advanced Topics in Software Engineering” in 2010. During the course, the students studied various domain engineering techniques, focusing on CBFM and ADOM and their ability to specify core assets. The study took place towards the end of the course as a class assignment. The students were equally divided into two groups of 9 students each according to their grades in relevant modeling courses, their academic and industrial background, and their familiarity with the examined methods. The students in the first group got a CBFM model of a domain and a dictionary of terms in that domain, while the students in the second group got an ADOM model of the same domain and the same dictionary of domain terms. The students in the two groups were asked to answer 15 true/false comprehension questions and to provide full explanations to their answers.

The results showed that CBFM outperformed ADOM in commonality-related questions, while ADOM outperformed CBFM in variability-related questions. This outcome is reasonable, as feature-orientation concentrates on a common kernel and its possible configurations, while ADOM treats software products and product lines as belonging to two different abstraction levels and allows more variability among products that belong to the same product line (e.g., via specialization and extension). Nevertheless, a statistical analysis showed that there is significant difference only in the variability specification and this is in favor of ADOM [19]. In all other categories no statistical significance was found. Still, according to the achieved averages, the overall comprehensibility in ADOM was better than that in CBFM. This outcome somehow questions the widespread opinion [20] that feature-orientation is simpler and, thus, more comprehensible to different stakeholders involved in SPLE and worth further investigation in the future.

5 Summary and Future Work

Different core assets modeling methods have been suggested. These methods are usually evaluated and compared subjectively, using different lists of criteria that highlight various aspects of core assets specification and utilization. We used a different approach for comparing these methods: empirical evaluation of comprehending and utilizing their resultant models. Based on a series of three studies, we noticed that variability is comprehensible and utilizable to a limited extent and that the main source of problem is in comprehending variation points. Yet, when providing explicit guidelines, the comprehension of the domain model increases.

The three conducted studies were relatively limited in their subjects' qualifications, the numbers of participating subjects, required tasks, examined methods, and provided models. Thus, in the future, we plan to replicate the empirical study on larger classes of trained domain engineering students and software developers and to use the suggested framework for comparing and evaluating core assets modeling methods in other categories, such as in Domain-Specific Languages [16].

References

1. Ahmed, F. and Capretz, L. F. The software product line architecture: An empirical investigation of key process activities. Information and Software Technology 50, pp. 1098–1113, 2008.
2. Bagheri. E. and Dasevic, G. Assessing the Maintainability of Software Product Line Feature Models using Structural Metrics. Software Quality Journal, Springer, DOI: 10.1007/s11219-010-9127-2, 2011.
3. Borba, C. and Silva, C. A Comparison of Goal-Oriented Approaches to Model Software Product Line Variability. ER'2009 workshops. LNCS 5833, pp. 244-253, 2009.
4. Chen, L. and Babar, M. A. A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology 53, pp. 344-362, 2011.
5. Clements, P. and Northrop, L. Software Product Lines: Practices and Patterns. Addison-Wesley, 2002.
6. Czarnecki, K. and Kim, C.H.P. Cardinality-Based Feature Modeling and Constraints: A Progress Report. In Proceedings of the OOPSLA Workshop on Software Factories, 2005.

7. Denger, C. and Kolb, R. Testing and Inspecting Reusable Product Line Components: First Empirical Results. Proceedings of the 2006 ACM/IEEE International Symposium on Empirical Software Engineering, ACM, pp. 184–193, 2006.
8. Djebbi, O. and Salinesi, C. Criteria for Comparing Requirements Variability Modeling Notations for Product Lines. The Fourth International Workshop on Comparative Evaluation in Requirements Engineering (CERE'06), in conjunction with RE'06, 2006.
9. Frakes, W.B. and Kyo, K. Software Reuse Research: Status and Future. IEEE Transactions on Software Engineering, 31 (7), pp. 529-536, 2005.
10. Gomaa, H. Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures, Addison-Wesley Professional, 2004.
11. Halmans, G. and Pohl, K. Communicating the Variability of a Software-Product Family to Customers. Software and Systems Modeling 2 (1), pp. 15-36, 2003.
12. Haugen, Ø. Møller-Pedersen, B., and Oldevik, J. Comparison of System Family Modeling Approaches. Software Product Lines Conference. LNCS 3714, pp. 102-112, 2005.
13. Kang, K., Cohen, S., Hess, J., Novak, W., and Peterson, A. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University, 1990.
14. Kyo, C. K. , Sajoong, K., Jaejoon, L., Kijoo, K., Euiseob, S. and Moonhang, H. FORM: A feature oriented reuse method with domain-specific reference architectures. Annals of Software Engineering 5 (1), pp. 143-168, 1998.
15. Matinlassi, M. Comparison of Software Product Line Architecture Design Methods: Comparison of Software Product Line Architecture Design Methods: COPA, FAST, FORM, KobrA and QADA. Proceedings of the 26th International Conference on Software Engineering (ICSE'04), 2004.
16. Mernik, M., Heering, J., and Sloane, A. M. When and How to Develop Domain-Specific Languages. ACM Computing Surveys (CSUR) 37 (4), pp. 316-344, 2005.
17. Ramesh, V. and Topi, H. Human Factors Research on Data Modeling: A Review of Prior Research, an Extended Framework and Future Research Directions, Journal of Database Management, Vol. 13 (2), pp. 3-19, 2002.
18. Reinhartz-Berger, I. and Sturm, A. Utilizing Domain Models for Application Design and Validation. Information and Software Technology, 51(8), pp. 1275-1289, 2009.
19. Reinhartz-Berger, I. and Tsoury, A. Experimenting with the Comprehension of Feature-Oriented and UML-Based Core Assets. Lecture Notes in Business Information Processing (LNBIP) 81, pp. 468–482, 2011.
20. Robak, S., Franczyk, B., Politowicz, K., Extending the UML for modeling variability for system families. International Journal of Applied Mathematics and Computer Science 12 (2), pp. 285-298, 2002.
21. Silva, C., Alencar, F., Araújo, J., Moreira, A., Castro, J. Tailoring an Aspectual Goal-Oriented Approach to Model Features. 20th International Conference on Software Engineering and Knowledge Engineering (SEKE'2008), pp. 472-477, 2008.
22. Sinnema, M. and Deelstra, S. Industrial validation of COVAMOF. Journal of Systems and Software 81 (4), pp. 584-600, 2008.
23. Svahnberg, M., Van Gurp, J., and Bosch, J. A Taxonomy of Variability Realization Techniques. Software – Practice & Experience 35 (8), pp. 705-754, 2005.
24. Trigaux, J.C. and Heymans, P. Modeling variability requirements in Software Product Lines: A comparative survey. Technical report PLENTY project, Institut d'Informatique FUNDP, Namur, Belgium, November 2003.
25. Webber, D. and Gomaa, H. Modeling variability in software product lines with variation point model. Science of Computer Programming, Vol. 53, pp. 305-331, 2004.
26. Ziadi, T., Hélouët, L., and Jézéquel, J.M. Towards a UML Profile for Software Product Lines. Software Product-Family Engineering (PFE'2004), LNCS 3014, pp. 129-139, 2004.

Knowledge Dimension in Business Process Modeling

Ligita Businska[1] and Marite Kirikova[2]

Faculty of Computer Science and Information Technology
Riga Technical Univesity, Latvia
{Ligita.Businska[1], marite.kirikova[2]}@cs.rtu.lv

Abstract. Business process models can be represented as stand-alone models and as a par-of a system of models. In the case of the system of models the business process model elements can be a part of other models that are included in the system of models. Each model that relates to the business process via its element can be regarded as a dimension of the business process. Thus the organizational structure model (performer model), goal model, decision model, location model, and other models represent a particular dimension of the business process. One of the dimensions that have not yet evolved into a model that could be easily related to the business process is knowledge dimension. The problem resides in the not fully agreed-upon understanding of the relationship between such notions as data, information, and knowledge. The concept of information code allows to look closer at knowledge dimension of the business process and to clarify several issues with respect to this dimension and its proper place in business process model representation.

Keywords: data, information, knowledge, business process model.

1 Introduction

The period of distrust in business process model based approaches due to unsuccessful re-engineering efforts in the previous century is over; and business process re-engineering again becomes an important topic in scientific literature [1-4]. However, it is worth to remember that business process engineering have to be a holistic approach and take into consideration various aspects of the business system, including organizational and individual knowledge [1-5]. In order to provide new means for the analysis of relationship between the business process and organizational knowledge we propose to include knowledge dimension in the business process model.

In business process modeling languages such as IDEF0, IDEF3, EPC diagrams in ARIS tool, GRAPES BM in GRADE tool, UML 2.0 activity diagram, and BPMN 2.0 data, information and material flow is often represented by the same symbols and without any unambiguous definitions of these concepts. On the other hand, knowledge modeling languages (KMDL, GPO-WM, PROMOTE, and RAD) allow to model knowledge, but do not address process logic to full extent and thus lose the possibility to represent data. Currently, from the point of view of various ways how data, information and knowledge are used in organizations, the following features of business process modeling languages are not yet fully supported in any of the above mentioned languages:

- Possibility to separate information and data during business process modeling
- Opportunity to identify the owner of data, information and knowledge
- Possibility to identify, plan, and manage knowledge of the role required for participating in a particular activity and linking this knowledge to competence model
- Possibility to evaluate the amount of lost organizational knowledge if a person – owner of knowledge – leaves the organization. i.e., to identify which tacit knowledge in this case should be transformed into explicit knowledge, such as documents, rules, systems, etc.
- Opportunity to improve understanding about the knowledge usefulness, validity and relevance for particular activities in a process
- Opportunity to enable competence requirements management and proactive training based on a process reengineering impact analysis.

We have already tried to address these issues with respect to BPMN notation in our previous work [6]. This lead to the introduction of specific symbols for data, information and knowledge objects. Experiments with the notation revealed that the relationship between the phenomena behind the symbols is somewhat unclear in the modeling process. Therefore in this paper we focus on analysis of this relationship by investigating intersection of modern information theory assumptions and knowledge management definitions of information and knowledge. The results obtained and their application for different business process modeling languages, as well as a template of activity representation with visible knowledge dimension are presented and discussed in this paper.

In Section 2 we ponder over the terms data, information, and knowledge and come to the conclusion that the use of information codes as a supplementary term helps to clarify relationship between previous three terms. We use all four terms to define information interaction in homogenous and heterogeneous environments. In section 3 we analyze information interaction in the context of business process modeling languages. In Section 4 the template of business process model activity with visible knowledge dimension and example of its use are represented. Section 5 consists of brief conclusions and points to the research for analysis of knowledge dimension of business processes.

2 Constituents of knowledge dimension

Data, information and knowledge are terms that are widely used, but still have no commonly agreed definitions. Data are usually associated to database, knowledge most often is associated to human beings while information is freely used in both cases. In this work we do not discuss various interpretations of the above mentioned terms deeply [7, 8, 9-11]. We focus on the relationship between data, information and knowledge and rely upon the following observations and assumptions:

1) Knowledge is located in the knowledge holder (natural or artificial)

2) Knowledge in the knowledge holder (e.g., human brain) has a particular structure which may be regarded as a “mental model”. The “mental model” can be natural or artificial, tacit and externalized, implicit and explicit
3) Any business process involves a knowledge process which is performed by a natural or artificial knowledge holder
4) If several knowledge holders are involved in the business process, - data, information, and knowledge exchange between them is possible. This exchange differs from the exchange of other substances as it is asymmetric: the amount of given information may differ from the received one; and the knowledge holder by giving information does not lose knowledge on the basis of which the information was provided.

To obtain a holistic and at least semi-formal view of the relationship between data, information, and knowledge we use theory that shows that in information exchange a substance called information codes is involved [7], i.e., information exchange is accomplished via information codes.

Suppose the knowledge holder (object O_1 provides some information codes T_1 to another knowledge holder (object O_2). The state transition in O_2 which receives this information is illustrated in Fig. 1. In the first phase, the object O_2 receives particular information code Ic_1. To perceive the code the object needs a particular “linguistic” device that can recognize the code. (E.g., if the code is information in English, it can be recognized if there is a “device” that can handle English). The received code is transformed into data Δd. Thus *data are functional values of information codes* which correspond to new parameters of object state obtained in interaction with another object.

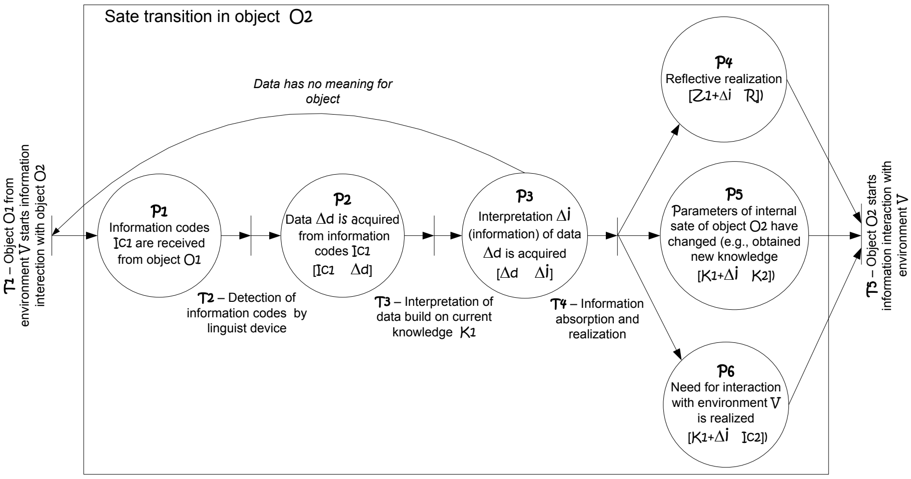

Sate transition in object O2
Data has no meaning for object
T1 – Object O1 from environment V starts information interection with object O2
P1 Information codes IC1 are received from object O1
T2 – Detection of information codes by linguist device
P2 Data Δd is acquired from information codes IC1 [IC1 Δd]
T3 – Interpretation of data build on current knowledge K1
P3 Interpretation Δi (information) of data Δd is acquired [Δd Δi]
T4 – Information absorption and realization
P4 Reflective realization [Z1+Δi R])
P5 Parameters of internal sate of object O2 have changed (e.g., obtained new knowledge [K1+Δi K2])
P6 Need for interaction with environment V is realized [K1+Δi IC2])
T5 – Object O2 starts information interaction with environment V

Fig. 1. State transition in knowledge owner when information codes are received.

In the next phase the object O_2 decides upon the meaning of obtained data Δd that is subjective interpretation of Δi by current knowledge of K_1 of O_2 taking into consideration M_1 – the set of its current needs or goals. According to [8] structured and processed data is information that is time dependent (relevant only in a given point of time) and correct with respect to the processed data set. In general, the amount of received information can be calculated as difference between knowledge after data interpretation and knowledge before interaction with object O_1: $\Delta i=Z_1-Z_2$. It can be regarded as a measure of reduction of uncertainty for choosing actions in order to achieve particular goals M_1 [12].

Information exists from the moment data is interpreted till the moment when the information has been absorbed or included in mental model of the object. As a result of information absorption the content or structure of mental model (including procedural and declarative knowledge which is stored in it) can be changed.

In the final phase realization of obtained information Δi takes place and it can lead to changes of internal state parameters of object O_2 or/and to the next cycle of interaction with environment. There can be several overlapping options of realization: (1) a reflective action: $K_1+\Delta i \rightarrow R$; (2) if the object starts the next cycle of iteration with object (-s) from its environment, object O_2 delivers appropriate set of information codes: $K_1+\Delta i \rightarrow Ic_2$: (3) if object changes its internal state, its mental model can change, under certain conditions obtaining new knowledge: $K_1+\Delta i \rightarrow K_2$. According to [13] *knowledge is reasoning about data* that is stored in object's "mental model" in order to promote action, problem solving, decision making, learning, and teaching. Knowledge is a higher organizational level of data that allows their specific interpretation. Requirements to data organization level can differ from a simple grouping of the data to complicated data hyper-structures.

Thus according to [7] a single cycle of information interaction between object and its environment is divided into three sequential phases: (1) object receives information codes from its environment, (2) obtained codes are interpreted, and finally (3) information is realized (reflected upon, absorbed, put into action). In Fig. 2 a simplified example with two objects (process performer that is analyst and document that includes interview protocols) is shown. The analyst performs the activity of analyzing as-is business process model. Perceived information codes are realized as new knowledge about actual business processes in the company.

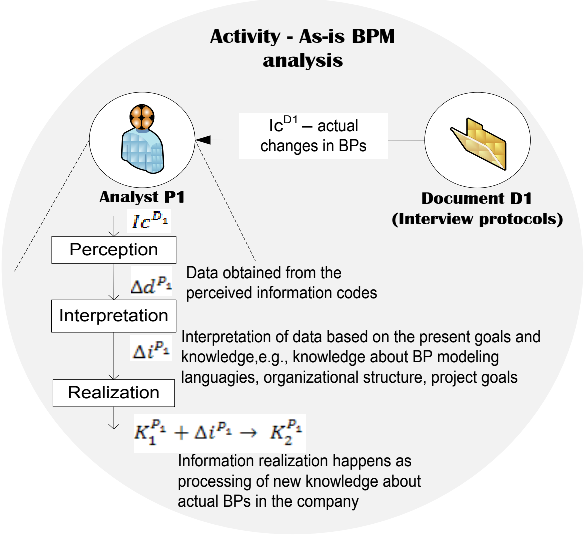

Activity - As-is BPM analysis
Ic^{D1} – actual changes in BPs
Analyst P1
Document D1 (Interview protocols)
Ic^{D_1}
Perception
Δd^{P_1}
Data obtained from the perceived information codes
Interpretation
Δi^{P_1}
Interpretation of data based on the present goals and knowledge,e.g., knowledge about BP modeling languagies, organizational structure, project goals
Realization
$K_1^{P_1} + \Delta i^{P_1} \to K_2^{P_1}$
Information realization happens as processing of new knowledge about actual BPs in the company

Fig.2. A simplified example of an activity.

The performer of a business process can receive information codes in three different ways, namely, from human, from active artificial object, and from passive artificial object. Depending on the situation the interchange of information codes can take place in homogenous (human-human, IS-IS) or heterogeneous (human-IS, IS-document, human-document) environments. In Fig. 3 and 4 internal changes of knowledge holders are illustrated. A in Fig. 3 shows information code interchange and new knowledge (natural or artificial) development in homogenous environment (on the left: human-human and on the right: computer system-computer system). Fig. 3 B illustrates how natural or artificial knowledge holder interacts with the passive knowledge holder (document). Figure 4 illustrates heterogeneous environment with two different types of knowledge holders. The interchange and knowledge development can proceed differently depending on the level of intelligence of the artificial knowledge holder (from the left to right: without data interpretation means; with data interpretation means only, and with learning ability).

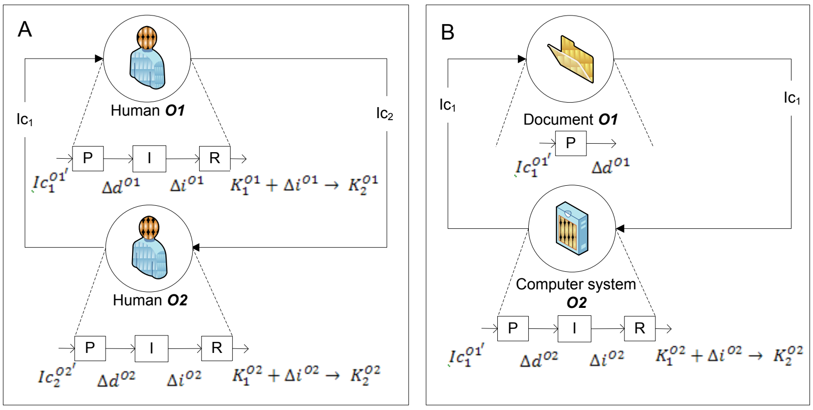


Fig. 3. A- Information interaction in homogenous environments; **B** - Information interaction between active knowledge holders and passive knowledge (P – perception, I – interpretation, R – realization).

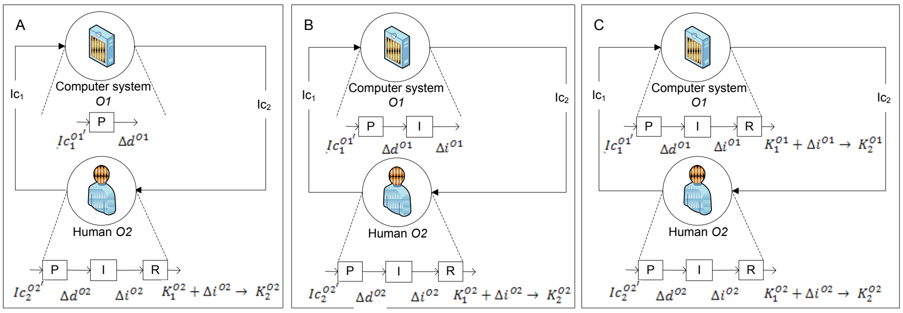


Fig.4. Information interaction in heterogeneous environment (among active objects) (P – perception, I – interpretation, R – realization).

The above analysis of information interaction shows that changes in knowledge are initiated by perception of particular information codes. Thus for representing the knowledge dimension it would be necessary to show knowledge before and after perception of information codes as well as coded information itself. The potential of contemporary business process modeling languages in this regard is examined in the next section.

3 Information exchange in business process context

In our previous work [6] we analyzed different attempts to include knowledge dimension in business process modeling and knowledge modeling languages and we proposed to integrate knowledge-oriented modeling language KMDL [14] and BPMN notation [15]. In that work three different objects: knowledge objects, information objects and data objects were used. However, further experiments with the integrated notation showed that it is difficult to distinguish between data and information objects. Theoretical issues discussed in the previous section clarify the reason behind this difficulty. It shows data rather as an internal than external phenomenon of the knowledge holder and interchange of perceivable knowledge is accomplished via information codes. None of the approaches analyzed in [6] took into consideration information codes and therefore are not directly applicable for representation of knowledge dimension in the way it is described in the previous section. On the other hand knowledge modeling approaches analyzed in [6] are not used very often; therefore in this work we consider "ordinary" business process modeling languages in order to see how appropriate they are for inclusion of knowledge dimension. The following business process modeling languages were analyzed: GRAPES BM – in GRADE tool [16], EPC diagrams in ARIS [17], IDEF 3 [18], IDEF 0 [19], UML 2.0 activity graphs [20], and BPMN 2.0 [15]. The languages were analyzed from the following two points of view (1) possibilities to represent data and knowledge (Table 1); (2) possibilities to represent process logics (Table 2). Both views are important for representation of static and dynamic aspects of knowledge in individual knowledge holders and in the process as a whole. In the Table 1 and 2 "-" means "does not support"; "-/+" means "somewhat supports"; "+" means "inclusion is possible"; "++" means "almost fully supports", and "+++" means "supports fully".

Table 1. Representation of inputs, outputs and resources

Criteria	GRAPES BM	ARIS EPC	IDEF 0	IDEF 3	UML 2.0	BPMN 2.0.
Input/output [data]	+	+++	+	-	+	++
Input/output [inforamtion]	+	+++	+	-	+	++
Input/output [knowledge]	-	+/-	-	-	-	-
Resource [knowledge]	-	-	-	-	-	-
Resource [human]	+	++	+	-	-	+
Resource [artificial]	+	+	+	-	-	+
Resource [data store]	+	+	-	-	+	-

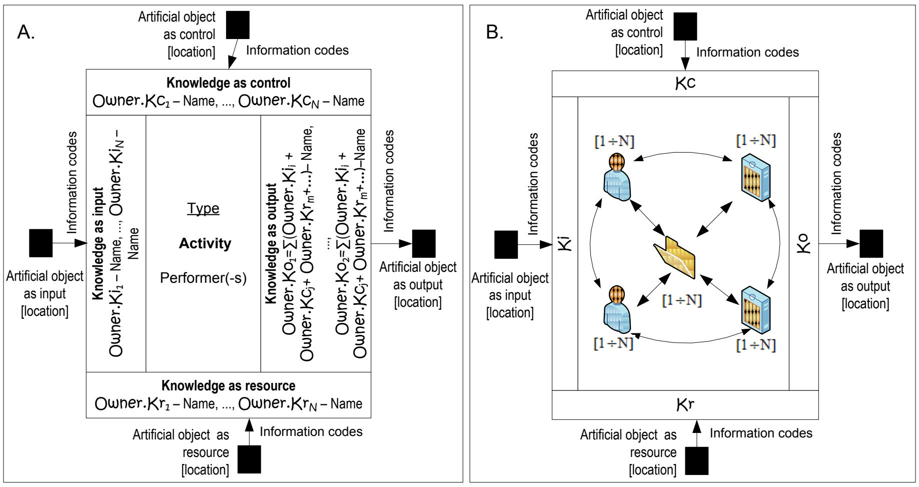
Table 2. Representation of process logics

Criteria	GRAPES	EPC	IDEF 0	IDEF 3	UML 2.0	BPMN
Process management	-/+	-/+	+	-	-/+	-/+
Controls	-/+	-/+	+	-	-/+	-/+
Decision points	+	+	-	-	+	+
Control flows	+	++	-	+++	++	+++
Events	+	++	-	+/-	+	+++

From the point of process logics the best options are BPMN and ARIS EPC. The least feasible is IDEF0, which lets to assume that this language has to be extended if taken as a basis for the representation of knowledge dimension.

4 Representing knowledge dimension transparently

In this section we propose one possible way how to represent an activity with knowledge dimension. We strive to show the proposed ideas graphically. It is not yet a new business process modeling notation. The representation is based on IDEF0 notation. IDEF0 was chosen as the basis for activity template, because it gives an opportunity to distinguish between controls (relates to knowledge holder's goals (see Section 2), inputs/outputs (received and produced information codes), and resources (knowledge in the holder). However, it must be admitted that IDEF0 notation is not the most suitable for representing logic of the process, therefore, in our further research we intend to combine it with other notations that give more means for control and decision points modeling. The activity template and example of its use are represented in Fig. 5 and 6.



A.
Artificial object as control [location]
Information codes
Knowledge as control
Owner.Kc₁ – Name, ..., Owner.Kc_N – Name
Information codes
Knowledge as input
Owner.Ki₁ – Name, ..., Owner.Ki_N – Name
Artificial object as input [location]
Type
Activity
Performer(-s)
Knowledge as output
Owner.Ko₁=Σ(Owner.Ki_i + Owner.Kc_j+ Owner.Kr_m+...)– Name,
...
Owner.Ko₂=Σ(Owner.Ki_i + Owner.Kc_j+ Owner.Kr_m+...)–Name
Information codes
Artificial object as output [location]
Knowledge as resource
Owner.Kr₁ – Name, ..., Owner.Kr_N – Name
Artificial object as resource [location]
Information codes
B.
Artificial object as control [location]
Information codes
Kc
Information codes
Ki
Artificial object as input [location]
[1÷N]
[1÷N]
[1÷N]
[1÷N]
[1÷N]
Ko
Information codes
Artificial object as output [location]
Kr
Artificial object as resource [location]
Information codes

Fig. 5. Activity with a knowledge dimension: A: activity template; B: activity zoomed in (this information is not presented in the template).

Each *Activity* (Fig. 5 A) corresponds to one of different combinations of interaction between human, computer systems, and documents as shown in Fig. 2-4. Social processes among performers inside the activity are not represented (Fig. 5 B). The activity template has the following attributes: *Activity name*, *Performers of the activity* (human or artificial (computer) system). For knowledge intensive activities there is an additional attribute *Type* with possible values Socialization, Externalization, Combination, and Internalization. These attributes and their values are visually positioned in the central part of the template. The central part is surrounded by four blocks that correspond to four types of knowledge, namely: control knowledge *Kc*, input knowledge *Ki*, output knowledge *Ko*, and Resource knowledge *Kr*. This is knowledge that is inside the knowledge holders (natural and/or artificial) participating in the activity and can be referred to as tacit knowledge. Each block of the tacit knowledge can be linked to particular artifacts: input artifacts *I*, output artifacts *O*, resource artifacts *R*, and control artifacts *C* which in essence are information codes perceived by tacit (natural or artificial) knowledge of the performers of the process. Each block *Kc*, *Ki*, *Ko*, *and Kr*, of the template can be related to particular concepts of the representation of organizational "mental model", if such is maintained. To illustrate the proposed template an activity of logical data model development process is illustrated (Fig. 6). The development process starts with an As-is business process model analysis when the analyst reads two documents: current business process model and an interview protocol. As a result of this activity the analyst should obtain new knowledge about actual business processes.

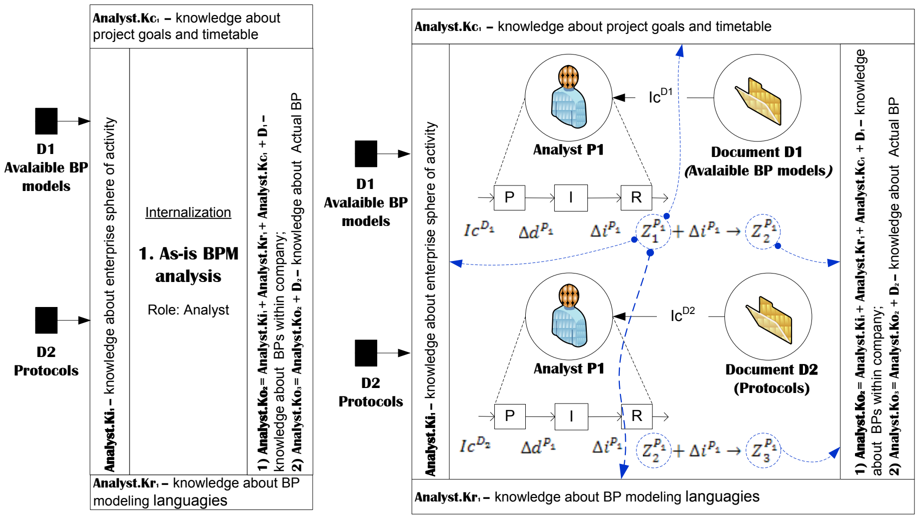

Analyst.Kc_1 – knowledge about project goals and timetable
D1 Avalaible BP models
D2 Protocols
Analyst.Ki_1 – knowledge about enterprise sphere of activity
Internalization
1. As-is BPM analysis
Role: Analyst
1) Analyst.Ko_2= Analyst.Ki_1 + Analyst.Kr_1 + Analyst.Kc_1 + D_1 – knowledge about BPs within company;
2) Analyst.Ko_3= Analyst.Ko_2 + D_2 – knowledge about Actual BP
Analyst.Kr_1 – knowledge about BP modeling languagies
Analyst P1
Document D1 (Avalaible BP models)
Document D2 (Protocols)
Ic^{D1}
Ic^{D2}
P
I
R
Ic^{D_1} Δd^{P_1} Δi^{P_1} $Z_1^{P_1} + \Delta i^{P_1} \rightarrow Z_2^{P_1}$
Ic^{D_2} Δd^{P_1} Δi^{P_1} $Z_2^{P_1} + \Delta i^{P_1} \rightarrow Z_3^{P_1}$

Fig. 6. Example of an activity represented by the template (on the left). What happens inside of it is illustrated on the right.

6 Conclusions

In business process reengineering it is important to have a holistic view of the enterprise. Since organizational knowledge is an essential aspect of an enterprise, there is a need for transparent linkage between the business process model and organizational and individual knowledge. In order to achieve this transparency the paper proposes a new activity template that gives visual means to relate business process to organizational knowledge and to analyze knowledge circulation in a business process. The model presented in the paper is in its experimental stage. Analysis of possibility to introduce it to different business process modeling languages is the next step of the research presented in the paper. Additionally we consider interviewing experts who routinely use business process modeling languages and notations in order to investigate how they presently capture knowledge.

References

1. Baimin B.S., Zijun H., and Xiohua G. Knowledge process reengineering and implementation of enterprise knowledge management. In International Conference on Information Management, Innovation Management and Industrial Engineering, 2010, pp. 23-26.
2. Wang B. Sh-B , Wang Ch , Yang J. Research on the reengineering of government business processes based on the environment of e-government. In: International Conference on E-Business and E-Government, 2010, pp. 4503-4506.
3. Li B.-Z., Liu Y. Organizational change pattern based on business process reengineering. In International Conference on E-Business and E-Government, 2010, pp. 1193-1197.
4. Chalaris I. E., Vlachopoulos S. Business Process Reengineering as a Modernizing Tool for the Public Administration- From Theory to Reality. In Fourth Balkan Conference in Informatics, 2009, pp. 64-69.
5. Weicher M, Chu W.W., Lin W. Ch., Le V., and Yu D. Business Process Reengineering: Analysis and Recommendations, available at http://www.netlib.com/bpr1.htm#reenghr
6. Supulniece I., Bušinska L., and Kirikova M. Towards extending BPMN with the knowledge dimension. in the Enterprise, Business-Process and Information Systems Modeling: Proceedings, Tunisa, Hammamet, 2010. - 69-81. lpp.
7. Янковский С. Я. Концепции общей теории информации. НиТ. Текущие публикации, 2001., avilable at http://n-t.ru/tp/ng/oti03.htm
8. Maier R. Knowledge management systems. Information and communication Technologies for knowledge management. Springer-Verlag Berlin Heidelberg, third edition, 2007.
9. Tiwana, A. Knowledge Management Toolkit, The: Practical Techniques for Building a Knowledge Management System, Pearson Education, 1999.
10. Beyon-Davies P.B. Significant threads: The nature of data, International Journal of Information Management 29 (2009) 170-188).
11. Francois Ch (Ed.) International Encyclopedia of Systems and Cybernetics, 2nd Edition, K.G. Saur, Munhen, 2004.
12. Corning P. A., Control Information Theory: 'The missing link' in the science of cybernetics, Systems research and behavioral science, Syst.Res. 24, 297-311 (2007)
13. Beckman, T. A methodology for knowledge management. Proceedings of the IASTED International Conference on Artificial Intelligence and Soft Computing (ASC'97), Banff, Canada, 1997, pp.29-32.
14. Gronau, N., Korf, R., Müller, C.: KMDL-Capturing, Analyzing and Improving Knowledge-Intensive Business Processes. Journal of Computer Science 4, pp. 452-472 (2005)
15. BPMN, available at http://www.omg.org/spec/BPMN/2.0/PDF
16. GRADE Business Modeling, Language Reference, Infologistik GmbH, 1998
17. ARIS Expert Paper, Business Process Design as the Basis for Compliance Management, Enterprise Architecture and Business Rules, March 2007
18. IDEF3, available at http://www.idef.com/IDEF3.html
19. IDEF0, available at http://www.idef.com/IDEF0.html
20. UML, available at http://www.visual-paradigm.com/VPGallery/diagrams/Activity.html