Selmin Nurcan
Elias Pimenidis

CAiSE Forum 2014

Proceedings of the CAiSE'2014 Forum at the 26th International Conference on Advanced Information Systems Engineering (CAiSE)

Thessaloniki, Greece, 16-20 June, 2014

CEUR -WS.org

Preface

CAISE is a well-established highly visible conference series on Information Systems Engineering. This year, the conference extends a special welcome to papers that address Information Systems Engineering in Times of Crisis.

The CAiSE 2014 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.

Three 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.

(3) Experience reports using the STARR (Situation, Task, Approach, Results, Reflection) template.

CAISE 2014 Forum has received 45 submissions from 29 countries (Algeria, Austria, Belgium, Canada, China, Colombia, Czech Republic, Estonia, France, Germany, Greece, India, Israel, Italy, Japan, Latvia, Luxembourg, The Netherlands, Norway, Portugal, Saudi Arabia, Singapore, Spain, Sweden, Switzerland, Tunisia, Turkey, United Arab Emirates, Uruguay). Among the submissions, 27 are visionary papers, 14 are demo papers and 4 are case study reports. 11 papers have been redirected from the main conference. 34 papers have been directly submitted to the Forum. The average acceptance rate of CAISE 2014 Forum is 35%.

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 Forum Program Committee and the Program Board. Eventually, 26 high quality papers have been selected; among them 16 visionary papers, a case study report and 9 demo papers.

The CAISE 2014 Pre-Proceedings available on this electronic support represent a collection of those 26 short research papers. Those papers included in the special proceedings issue titled "CAiSE 2014 Forum" are published by CEUR.

We would like to express our gratitude to the members of Forum Program Board and the Forum Program Committee for their efforts in providing very thorough evaluations of the submitted Forum papers. We wish also to thank all authors who submitted papers to the Forum for having shared their work with us.

After CAiSE 2014, Forum authors will be invited to submit an extended version of their papers, enriched thanks to the fruitful discussions during the CAISE Forum, for Forum post-proceedings that will be published as a Springer LNBIP volume.

Last but not least, we would like to thank the CAISE 2014 Program Committee Chairs and the Local Organisation Committee for their support.

June 12th, 2014
Selmin Nurcan, Elias Pimenidis
CAISE 2014 Forum Chairs

Program Board Members

Xavier Franch, Universitat Politecnica de Catalunya, Spain
Marite Kirikova, Riga Technical University, Latvia
Barbara Weber, University of Innsbruck, Austria
Jelena Zdravkovic, Stockholm University, Sweden

Program Committee Members

João Paulo A. Almeida, Federal University of Espírito Santo, Brazil
Said Assar, Telecom Ecole de Management
Luciano Baresi, Politecnico di Milano, Italy
Judith Barrios, Universidad de Los Andes, Venezuela
Kahina Bessai, University Paris 1 Panthéon Sorbonne, France
François Charoy, Nancy-Université, France
Fabiano Dalpiaz, Utrecht University, The Netherlands
Maya Daneva, University of Twente, The Netherlands
Sergio España, Universidad Politécnica de Valencia, Spain
Christos Georgiadis, University of Macedonia, Greece
Johny Ghattas, University of Haifa, Israel
Aditya Ghose, University of Wollongong, Australia
Chihab Hanachi, University Toulouse 1 Sciences Sociales, France
Charlotte Hug, University Paris 1 Panthéon Sorbonne, France
Evangelia Kavakli, University of the Aegean, Greece
Agnes Koschmider, Karlsruhe Institute of Technology, Germany
Elena Kornyshova, CNAM, France
Hui Ma, Victoria University of Wellington, New Zealand
Raimundas Matulevicius, University of Tartu, Estonia
Sai Peck Lee, University of Malaya, Kuala Lumpur, Malaysia
Naveen Prakash, GCET, India
Hajo Reijers, Eindhoven University of Technology, The Netherlands
Gustavo Rossi, Universidad Nacional de La Plata, Argentina
Motoshi Saeki, Tokyo Institute of Technology, Japan
Samira Si-Said Cherfi, CNAM, France
Jacques Simonin, Télécom Bretagne, France
Guttorm Sindre, NTNU, Norway
Janis Stirna, University of Stockholm, Sweden
Arnon Sturm, Ben-Gurion University, Israel

Information Systems and Performance Management for Collaborative Enterprises: a proposal

Barbara Livieri[1], Mario Bochicchio[2]

[1]Department of Economic Sciences
[2]Department of Engineering
University of Salento, Lecce, Italy
{barbara.livieri,mario.bochicchio}@unisalento.it

Abstract. Much of the research on performance management (PM) for collaborative enterprises (CE) is based on qualitative considerations and does not consider the impact of modern Information Systems both on the collaborative/competitive dimension of firms and on the PM process. The peculiarities of the different types of CEs are not clearly addressed and managed and the performance measurements are often oriented to specific aspects rather than to assess the overall quality of business. Moreover, in several proposals, the skills and the time required to the managers of CEs are far from those available in the largest part of existing SMEs. In this scenario the objective of the paper is to discuss how enterprise modeling techniques can contribute to enhance the governance of collaborative enterprises.

Keywords: Information Systems; Enterprise Modelling; Performance Evaluation; Collaborative Enterprises; Ontologies.

1 Introduction

Strategic alliances, virtual organizations and other forms of Collaborative Enterprises (CE) are gaining ever more importance due to globalization, which has forced businesses to rearrange their organizational structures. In the last twenty years, organizational relationships have moved from intra-organizational to inter-organizational ones and are moving towards trans-organizational relations, with a prediction of a speed for value creation never seen before [1]. Nonetheless, it is known that globally between 50% and 70% of CEs fails [2, 3], often due to the lack of a comprehensive analysis that combine strategic goals and KPIs, whereas performance measurement is a key element in turning goals into reality [4]. In fact, although several authors [5] studied the role of management accounting in inter-organizational environments, to our knowledge no one applied these results in order to quantitatively analyze the performance of CEs, of involved firms and of their linkage [1, 6, 7] for CEs governance purposes. Moreover, in several proposals, the skills required for CEs' managers are far from those available in the largest part of existing SMEs, which are the most numerous actors in CEs. In this context, firms would benefit from methodologies and tools allowing them to better link desired objectives and achieved results in an inter-organizational environment. This requires a more structured and systematic approach to evaluate not only the firms' own performance but also how it compares with partners and competitors [8], even in different CEs. In practical cases, this kind of interrelated performance evaluation and comparison can't be conceived and realized without a set of suitable IS elements and procedures, which becomes not neutral with respect to the measured performance and to the style of management adopted for modern CEs, as well as a music instrument is not neutral with respect to the played music. In this perspective, Information Systems (IS) have to face the new challenge offered by networked enterprises [9, 10] and IT (Information Technology) concepts become relevant to CEs for doing business: (a) online databases and information modeling assume a key role in managing information and in exchanging it with stakeholders; (b) workflow systems and process modeling become essential to understand how firms and CEs are structured, how they interact with each other's (inter-organizational process modeling), and how they can react to external stimuli; (c) ontologies and Semantic Web techniques become necessary to manage the increasing amount of knowledge and documents (contracts, benchmarks, products, services ...) flowing in CEs; (d) Cloud computing, big data, business analytics, advanced computation and visualization techniques can push CEs to a new levels of understanding about business. In order to start exploring the implications of these assumptions we decided: a) to analyze the existing literature on performance measurement and IS in the perspective of a theoretical foundation for performance monitoring in collaborative enterprises enabled by online IS; b) to elicitate a set of requirements, starting from the gaps in existing literature and from stakeholders' goals; c) to propose an approach that can satisfy these requirements.

The paper is organized as follows: first, a literature review is presented to define a foundation for the explorative research; second, the method is described and a conceptual framework is proposed to organize both the elements coming from the literature and the first evidences coming from the field; third, the main concepts from enterprise modelling and others IS-related research areas are analyzed in relation to performance-monitoring for CEs. Last section is for concluding remarks and recommendation for future research.

2 Literature analysis and theoretical foundation

In this paragraph we will analyze the literature on performance measurement for CEs, in order to understand the domain of application, and on cross-organizational IS to define the research problem. For each topic, we will outline the current state of research and the existing gap and we will analyze the prospective of future research, thus how these topics have to evolve in order to face the new challenges deriving from the changes in society. Finally, we analyze related works on enterprise modeling.

2.1 Background and research problem outline

First, performance management and performance measurement have a key role in the assessment of CEs and of how the CE is affecting firms. However, enforcement methods, such as Open Book Accounting (OBA), which allows firms to share accounting information, are sometimes seen as formal control mechanism that damages trust [11], and there are still few works on how to measure the effects of CEs on firms [12], and even in those there is no focus on quantitative aspects [1, 6]. Also, there are few works that take into account both CEs and SMEs [13]. Therefore, whilst we are going towards a network-SMEs-driven society, new challenges arise for performance measurement systems, since they have to be developed and used across the traditional organizational boundaries. The question is how to manage both the performance of CEs and of firms for SMEs [1]: it is necessary to modify existing tools for inter-organizational settings, overcoming the clear-cut between external and internal environment. Indeed, whilst it is possible to use the same performance measurement frameworks used for firms, it is still necessary to structurally and operatively change the measurement system [14].

Another relevant aspect concerns cross-organizational Information Systems (IS), which can assure a flow of information among and within organizations [15], thus coordination among partners, which is a key factor in order to achieve goals. However, according to the contingency theory, a change in the organizational structure implies a change in the IS. In this sense, IS usually distinguish and oppose relations within a firm, from those across it, whilst in an inter-organizational setting it is necessary to broaden data sources so to include partners and to consider them as beneficiary of the information [16]. At the Enterprise Systems level, this can be achieved through shared databases, data warehouses, workflow management systems, web services, SOAs or cross-organizational ERP, which are used from several independent firms whom cooperate in an inter-organizational environment (value web) [17]. The use of cross-organizational ERP systems can lead to a lost on flexibility because it implies processes standardization and collaborative relations are not always stable. Anyway, most of the IS adopted are not cross-organizational; thus, "they focus on a single enterprise with some supports towards sharing performance information with external parties" [1]. However, the key element in the future seems to be "cooperation" [10], whilst IS should "enable new forms of participation and collaboration, catalyze further the formation of networked enterprises and business ecosystems […] ushering in a new generation of enterprise systems" [9]. Therefore, the question is how to design and develop IS for CEs and for networked SMEs, allowing a monitoring at two levels of granularity: the CE level and the firm level, with a guarantee of comparability between KPIs and perspectives of the two levels.

2.2 Related works

Part of the literature on enterprise modeling concerns performance indicators, which are modeled by means of domain-specific modeling languages (DSML) and ontologies. The aim is to offer models able to support the creation and the effective and efficient interpretation of "*performance measurement systems* […] *by providing differentiated semantics of dedicated modeling concepts and corresponding descriptive graphical symbols*" [18]. In particular, Popova and Sharpanskykh [4] developed a framework for modeling KPIs and their relations through dedicated first-order sorted predicate logic-based modeling language, while temporal relations are expressed through Temporal Trace Language (TTL). With similar intentions in [19] the Business Intelligence Model

(BIM) is used in order to model the strategy and the related goals, indicators and potential situations (Strengths, Weaknesses, Threats and Opportunities). In [20] techniques and algorithms to define KPIs metrics expression and value are developed. Moreover, In [18] a model for enabling reflective performance measurement, namely MetricM, and a domain-specific modeling language, named MetricML, are offered. Even though these works, and in particular [18], offer a broad analysis of performance indicators and of their relations, DSML and semi-formal frameworks cannot be directly integrated in IS. In this sense, in [21], the authors develop an ontological approach for the definition of Process Performance Indicators (PPIs) through OWL DL. However, only PPIs are taken into account, without perhaps considering the relation between goals and KPIs. A wider range of indicators and the analysis of the related objectives would be indeed useful to assess the overall performance of the firm. An interesting work has been done in [22] where an ontology of KPIs with reasoning functionalities for Virtual Enterprises is presented. The model enables the definition and manipulation of heterogeneous KPIs calculated in partner firms. The main reasoning functionalities are formula manipulation, equivalence checking, consistency checking and extraction of common indicators. Nonetheless, authors put much of their focus on innovation processes and not on the firm as a whole. Also, in their model they don't consider goals. In general, there are still few works that analyze ontologies of KPIs and a lack of works that simultaneously take into account KPIs, goals and CEs, which are entities far more complex than individual enterprises.

3 Method

As recommended by [9], the development of IS can't follow anymore a technology-driven approach, but has to follow a technology-enabled enterprise-driven approach. Therefore, in order to develop our approach for the design of IS elements for CEs, we used KAOS [23], a goal-oriented approach coming from requirements engineering. In particular, in order to elicitate requirements, we took into account both stakeholders goals and gaps in existing literature (Section2). After analyzing about 200 peer-reviewed research papers on CEs and after interviewing about 20 people directly involved in CE management, we identified seven main stakeholders interested in CEs and in measuring their performance: firms' and CEs' managers, researchers interested in CEs, network associations, policy makers, banks and consultants. For all of them we defined a complete set of goals and constraints ordered by priority, taking into account:

- their role along the CE's lifecycle, since not all stakeholders have a specific interest in all phases. E.g., while firms have an active role in all phases, CEs' managers are interested only in operational and conclusion phase. Therefore, we distinguished goals in a 'pre-alliance', 'operational life' and 'conclusion of CEs life' ones.
- information granularity since each stakeholder needs for information at different levels of granularity, where the 'elementary information' relevant to the stakeholders is about the performance indicators of each firm. E.g., firms' managers need detailed information on their firm and partners and competitors, CEs' manager details on the

CE and more synthetic information on firms, policy makers synthetic information on the whole system of CEs.

Other dimensions of analysis can be added, but an important point is that all stakeholders characterized by similar values of dimensions can be associated to a similar set of goals, requirements and constraints. As exemplification, a set of goals should include:

G.1 in the pre-alliance phase, each firm is interested in evaluating the suitability of collaboration for the achievement of specific objectives (e.g., growth in R&D);

G.2 in the alliance-operative phase, each CE manager is interested in analyzing their own KPIs and of other CE' related data;

G.3 in all phases, CEs' managers are interested in benchmarking, performed by comparing their KPIs with those of CEs with homogeneous characteristics;

G.4 in all phases, firms want to control what to show (nothing, just minimal data, financial ratios, etc.) to others users (CEs' partners, external observers, etc.).

From these goals and literature, we defined the following requirements for an IS designed for collaborative SMEs:

- **Requirement 1**. *Define a shared language for KPIs*. Indeed, KPIs can be calculated or interpreted in several ways, making them not comparable within or among CEs.
- **Requirement 2**. *Have a comprehensive analysis of the phenomenon, taking into account CEs type, lifecycle, organizational structures, roles and goals*. CEs are heterogeneous clusters of partnerships among enterprises (FInES 2012): therefore, for benchmarking purposes, it is obviously not enough to compare CEs only taking into account the business sector or the size, since other factors come into play.
- **Requirement 3**. *Build domain-specific KPIs, i.e., specific for the CE type, maturity and goals*. Different CEs need for different KPIs [25]; therefore firms and CEs have to understand which KPIs are relevant and what a KPIs mean in a given firm or a CE with defined goals. This kind of understanding is not immediate, especially in several SMEs, which lack of the know-how needed to perform this kind of analysis.
- **Requirement 4**. *Provide graphical representations, in order to reduce the complexity of the analysis and of the monitoring of CEs performance*. CEs are a multifaceted phenomenon, difficult to analyze and to comprehend in abstract ways. The mere analysis of CEs' goals, type and related KPIs could be misleading for managers.
- **Requirement 5.** *Assist in the contract drawing and enactment*;
- **Requirement 6.** *Guarantee privacy*.

4 Conceptual framework

A CE can be seen as a system [25] composed by three layers: the alliance layer, the firm layer and the relation (among firms and between each firm and the CE) layer. For each layer we can create an information repository: the upper layer (alliance layer) is for information on the CE coming from several data sources (financial statements, web sites and so on). This information regards e.g., objectives, activities, results achieved, and the program. The lower layer (firm), is for information on firms participating in CEs, coming from several data sources (financial statements, web sites and so on). This information concerns objectives, activities, business sector, characteristics, organizational structure and performance. Finally, the middle layer (relation/formal or informal agreement) is for information on contracts, governance and duration of the collaborative enterprise. The analysis of repositories enable the creation of a database for CEs and firms, with a list of collaborations and objectives. This can facilitate the search for partners (firms or CEs), thus supporting and simplifying the partner selection process. The repository of the relation layer also allows for the storage of contracts, whereas available. In this frame, ontologies have a double role. First, we can provide a semantic representation of the information on the repositories, with a classification of CEs along three dimensions, namely CE type, maturity and objectives. Crossing the three dimensions of analysis enables the construction, by means of reasoning functionalities, of reflective [18] domain-specific KPIs, i.e., KPIs specific for the type of alliance, the maturity and the objectives. For example, some of the domain specific KPIs for a supply chain with an informal-technical based connection at the early stages of the CE and with the goal of cost reduction are the following:

- Overall production costs variation between t_0 (before-CE) and t_1 (after-CE), since the comparison between two periods of time is an effective indicator [14];
- Overall transportation costs variation between t_0 and t_1;

Moreover, ontologies can be applied on the contract and organizational repository in order to provide domain-specific contracts and organizational structures templates, such as those provided by the Legal-IST project (www.legal-ist.org), for firms that decide to formalize the collaboration. This approach enables the representation of the linkage between alliances' and firms' goals and KPIs and makes possible to track which KPIs are used from firms with specific goals, of a specific type and with a certain maturity, so that this information are stored and used to suggest to not expert users which KPIs to choose. In short, this approach can facilitate firms also in the choice of which KPIs to include in the dashboard, thus which KPIs are relevant for their goals, CE type and maturity. Indeed, through data visualization tools and KPIs ontologies it is possible to develop an interpretative framework able to understand KPIs and to offer information on relevant variables, depending on the typology of partnership. This is particularly useful for SMEs, who lack of the skills to develop performance measurement systems.

These features can be offered through a collaborative, cloud-based Information System. As stated in [1], IS are essential for the development and use of Performance Measurement Systems. Moreover, the IS has to operate in an inter-organizational setting, thus it has to be Interned-based in order to be easily accessible by all firms. Also, when SMEs come into play, it is important to use Clouds to permit a suitable scalability and low costs: with a unified system the costs for the development and maintenance of the IS are shared costs, thus firms and CEs can use IS with little investments. Furthermore, the IS system should allow firms and CEs monitoring, through the creation of personalized dashboards, elaborated through reasoning features and queries on the ontology, KPIs evaluation and information sharing. Monitoring techniques should be integrated with benchmarking features as well, through which it's possible to compare firms or CEs with similar ones, without the necessity to provide analytic data on costs and revenues and, thus, overcoming one of the main limits of management accounting solutions such as open book accounting (OBA). Finally, in the collaborative IS firms should be able to share information, in order to better collaborate with partners and to have more detailed benchmarks, with different level of privacy. This means that each firm can be a grey box, a white box or a black box for each other firm. In more detail, it is a) a white box if choose to be completely transparent for other firms, e.g., disclosing its processes and organizational structures, b) a black box if the firm choose to disclose to other firms only external parameters (e.g., financial statements, information on web sites); c) a gray box if the firm choose to disclose only partial information.

The proposed approach can overcome the clear cut between external and internal environment since such a service, partially based on ontologies which enable a shared knowledge of the domain, should allow the creation of aggregated performance, without the need to disclose the atomic values. Also, the use of benchmarking techniques can overcome the issue of implementing highly-complex performance measurement systems, which are too expensive in terms of financial and organizational resources.

5 Discussion and conclusions

In this paper, through the analysis of existing literature, we discussed how the research on Information Systems (IS) can contribute to reshape the PM process to better integrate it in the management cycle. In this perspective, Information Systems (IS) have to face the new challenge offered by a networked society. Starting from the literature analysis we elicitate a set of requirement and propose an approach for the development of a comprehensive service, based on enterprise modelling techniques, for CEs governance and analysis, through the creation of a collaborative IS and of repositories, and the use of ontologies. With respects to related works on DSML for performance measurement, we use ontologies, which can be easily integrated in IS or online services; on the other hand, the ontologies proposed in literature don't consider jointly the inter-organizational settings and the linkage between KPIs and goals. In particular, in the present work, we developed a reference framework useful for understanding KPIs in relation to CEs goals, types and maturity signaling promptly anomalies and offering information on relevant variables, depending on the typology of CEs. The application of this approach is particularly useful when SMEs comes into play, since they often lack of the financial and managerial resources required to enforce a complex and heterogeneous performance measurement system. Future research should move towards the development of cloud based IS designed for collaboration among SMEs.

6 References

1. Bititci, U., Garengo, P., Dörfler, V., Nudurupati, S.: Performance Measurement: Challenges for Tomorrow. Int. J. Manag. Rev. 14, 305–327 (2012).
2. Kaplan, R.S., Norton, D.P., Rugelsjoen, B.: Managing Alliances with the Balanced Scorecard. Harv. Bus. Rev. 114–121 (2010).
3. Bititci, U.S., Butler, P., Cahill, W., Kearney, D.: Collaboration: a key competence for competing in the 21st century, (2008).

4. Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. Inf. Syst. 35, 505–527 (2010).
5. Caglio, A., Ditillo, A.: A review and discussion of management control in inter-firm relationships: Achievements and future directions. Elsevier Ltd (2008).
6. Bochicchio, M.A., Livieri, B., Longo, A., Di Cagno, P.: DsNA: a Database for Strategic Network Analysis in Italy. itAIS 2013 (2013).
7. Krathu, W. et al.: Identifying Inter-organizational Key Performance Indicators from EDIFACT Messages. 2013 IEEE 15th CBI. pp. 276–283. IEEE (2013).
8. Parmenter, D.: Key Performance Indicators: Developing, Implementing,and Using Winning KPIs. Wiley (2011).
9. FInES Research Roadmap Task Force: Future Internet Enterprise Systems (FInES)-Research Roadmap. (2010).
10. Missikoff, M.: The future of enterprise systems in a fully networked society. Advanced Information Systems Engineering. pp. 1–18. Springer (2012).
11. Windolph, M., Moeller, K.: Open-book accounting: Reason for failure of inter-firm cooperation? Manag. Account. Res. 23, 47–60 (2012).
12. Dekker, H.C.: Value chain analysis in interfirm relationships: a field study. Manag. Account. Res. 14, 1–23 (2003).
13. Pittino, D., Angela, P., Mazzurana, M.: Alliance Governance and Performance in SMEs : Matching Relational and Contractual Governance with Alliance Goals. 3, 62–83 (2013).
14. Bititci, U.S., Martinez, V., Albores, P., Parung, J.: Creating and managing value in collaborative networks. Int. J. Phys. Distrib. Logist. Manag. 34, 251–268 (2004).
15. Eckartz, S., Katsma, C., Daneva, M.: The Inter-organizational Business Case in ES Implementations : Exploring the Impact of Coordination Structures and Their Properties. Enterp. Inf. Syst. Int. Conf. CENTERIS 2010. (2010).
16. Håkansson, H., Lind, J.: Accounting and network coordination. Accounting, Organ. Soc. 29, 51–72 (2004).
17. Daneva, M., Wieringa, R.: Cost estimation for cross-organizational ERP projects: research perspectives. Softw. Qual. J. 16, 459–481 (2008).
18. Strecker, S., Frank, U., Heise, D., Kattenstroth, H.: MetricM: a modeling method in support of the reflective design and use of performance measurement systems. Inf. Syst. E-bus. Manag. 10, 241–276 (2011).
19. Mat, A., Trujillo, J., Mylopoulos, J.: Conceptualizing and Specifying Key Performance Indicators in Business Strategy Models. 282–291 (2012).
20. Barone, D., Jiang, L., Amyot, D., Mylopoulos, J.: Composite indicators for business intelligence. Conceptual Modeling–ER 2011. pp. 448–458. Springer (2011).
21. Resinas, M., Ruiz-cort, A.: Defining Process Performance Indicators : An Ontological Approach. 555–572 (2010).
22. Diamantini, C., Potena, D., Storti, E.: A Logic-Based Formalization of KPIs for Virtual Enterprises. Advanced Information Systems Engineering Workshops. pp. 274–285. Springer (2013).
23. Dardenne, A.: Goal-directed acquisition. 20, 3–50 (1993).
24. FInES Task Force on Collaborative Networks and SOCOLNET: Taxonomy of Collaborative Networks Forms. (2012).
25. Parung, J., Bititci, U.S.: A conceptual metric for managing collaborative networks. J. Model. Manag. 1, 116–136 (2006).

Understanding Service Variability for Profitable Software as a Service: Service Providers' Perspective

Eng Lieh Ouh[1] and Stan Jarzabek[2]

[1] Institute of Systems Science,
National University of Singapore
25, Heng Mui Keng Terrace, Singapore 119615
englieh@nus.edu.sg
[2] Department of Computer Science, School of Computing,
National University of Singapore
Computing 1, 13 Computing Link, Singapore 117417
dcssj@nus.edu.sg

Abstract. The number of tenants that subscribe to and pay for a service, and the cost of the SaaS computing infrastructure are the main factors that drive service profitability. In many application domains tenants' requirements for service vary. Service variability is the degree of the actual variability provided by the Service Provider over the variability required by the tenants for the service. With growing number of tenants, the likelihood of facing more diverse tenants' requirements increases. We conducted a study of how choices regarding service architecture affect service variability and the cost of supporting a service. We identified positive and negative impacts of service architectural choices on service variability and on Service Provider's costs. We illustrated how the knowledge of those impacts can help Service Providers analyze service profitability based on different service architecture models, leading to more-informed decisions regarding adoption of SaaS.

Keywords: Software as a Service, SaaS, service variability, service architecture, Profitability

1 Introduction

Service variability is the degree to which Service Provider can accommodate tenant-specific requirements into a service. The Service Provider tries to accommodate these requirement variations into the service so as to better fit the service to the tenants. As the unfit costs for the tenant decreases, the relative economic advantage of the SaaS business model increases [1]. If tenants' requirements for service vary only moderately, it is possible to engineer required variability into a service on cost-optimal (from Service Provider perspective) SaaS architectural model whereby all the tenants share the same service instance during

service execution. Dynamic binding techniques may be sufficient to address modest variations in service requirements. However, such cost-optimal SaaS solution may not be feasible if tenants' requirements differ in more drastic way. Shared service instance and dynamic binding techniques impose limits on how far we can vary service requirements. Then, a Service provider might consider SaaS architectural model with dedicated service instance for each tenant. Operational cost of such architecture is higher than that of shared instance, but dedicated instance architecture opens much more powerful options for engineering high-variability, adaptable services with static binding techniques.
To come up with a SaaS solution that maximizes profits, a Service Provider must weigh the revenue from selling a service to potentially many tenants, against the cost of SaaS computing infrastructure to support the service. Given inter-dependencies among factors that collectively determine profitability of service offering, the task is not easy.
We conducted a study of how choices regarding service architecture affect service variability and the cost of supporting a service. We identified positive and negative impacts of service architectural choices on service variability and on Service Provider's costs. We illustrated how the knowledge of those impacts can help Service Providers analyse service profitability based on different service architecture models, leading to more-informed decisions regarding adoption of SaaS. Our study is qualitative. In future work, we will extend it with quantitative analysis and models that more precisely correlate SaaS costs and benefits, giving more accurate insights into profitability of SaaS from service variability perspective.
The paper is organized as follows: We first describes the architectural choices of SaaS relevant to service variability in Section 2. In Section 3, we introduce the architectural models and further analysis the scenarios related to service variability in Section 4. Section 5 is on related work and Section 6 is our conclusion.

2 Techniques and Saas Architectural Choices Relevant to Service Variability

2.1 Service Engineering

1. *Static Binding Variability Techniques (SBVT)* - Static binding techniques instrument service code for adaptability to tenants' variant requirements at the design time. During (pre-)compilation or build time, variant requirements are bound to the variation points in service code to produce a custom service. Commonly used variation techniques include preprocessing (macros), Java conditional compilation, commenting out feature code, design patterns , templates and parametrisation, and build tools (e.g., make or Ant). XVCL [2] extends the concept of macros to provide better support for variability management in terms of generic design and separation of concerns.
2. *Dynamic Binding Variability Techniques (DBVT)* - Using dynamic binding techniques, we design a service that can adapt itself to the needs of

different tenants at runtime. Design patterns, reflection and parameter configuration files consulted during service hosting exemplify dynamic binding techniques. One common technique is using Aspect-Oriented Programming which involves specifying of aspects point cuts and advices that will describe the variability. Another common technique is Service Oriented Architecture Service Binding and Registry Lookup which involves registering of variants in the registry. Application components lookup the registry at runtime to dynamically bind variants to a service.

2.2 Service Packaging

As new tenants are on-boarded and requirements of existing tenants or service functions change, service must be adapted to accommodate evolving needs of tenants. Service adaptation has to be done without affecting existing tenants.

1. *Service Level Encapsulation (SLE)* - For Service Level Encapsulation, a service is implemented with identified shared service components. There is clear separation of components for each service, but not between tenants. The tenants who are using the service can be temporary affected during service modification but the tenants who are not using the service will not be affected.
2. *Tenant Level Encapsulation (TLE)* - For Tenant Level Encapsulation, a service is implemented with specific service components for each tenant. There is clear separation of components for each tenant. During service modification, only the specific tenants are affected.

2.3 Service Hosting

A service can be hosted on single or multiple application instances. Application instance is a software process (executable application code) running on an infrastructure platform.

1. *Shared Instance (SI)* - For Shared Instance, tenants access a service through a common application instance. The Service Provider considers this option typically to maximize resource utilization.
2. *Dedicated Instance (DI)* - For Dedicated Instance, each tenant accesses the service on its own dedicated application instance. The Service Provider considers this option due to high variation in tenants' requirements or for compliance to service level agreements.

A summary of the architectural choices is shown in a tree structure in Fig. 1.

2.4 Impact of the Architectural Choices

The architectural choices for service variability impact the degree, costs and benefits of the service variability. Table 1 summarize these relationships.

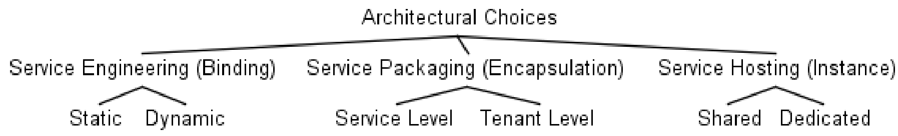


Fig. 1. Architectural choices

1. *Degree of Service Variability* - The degree of service variability is the extent and scope to which variations in service requirements can be handled. To understand the possible variations in service requirements, we use the entity-controller-boundary pattern to break down each service into its service components. Each service can be represented by interactions among a set of service components in terms of boundary, controller and entity components. Boundary components are used to interface externally with information elements and can varies with the elements and its representation termed as *Interface Variability.* Controller components manage the interactions among components. The composition of components including the flow of interactions and processing logic can varies among tenants result in *Composition Variability* and *Logic Variability.* Entity components represent the data of the service and can varies in the type of data elements and its structures. We termed this as *Data Variability.* The variations in service requirements for each service is the sum of all the variations of its service components for that service. A service is able to achieve high degree of service variability if it has the ability to handle high variations of service requirements.
2. *Costs and Benefit of (Profitability of investing in) Service Variability* - The cost incurred includes the cost for designing or re-designing the service binding in service engineering and service packaging. It also includes the infrastructure cost for service hosting to support service variability. High degree of service variability can be better supported by static binding (service engineering), dedicated instance (service hosting) and tenant level encapsulation (service packaging). However, these decisions requires higher costs in terms of design efforts and computing resources to run the service. The benefit of managing service variability is to increase the revenue by widening the tenants base. By being able to support higher degree of service variability, the tenant base can be increase easily. In a given situation, the Service Provider needs to make decisions to minimize the cost of service engineering/service packaging/service hosting and maximize the revenue (by widening the tenant base), ultimately affecting the profitability of a service.

3 SaaS Architectural Models

SaaS Architectural Models differ in how the service code is managed during service engineering, service execution and service hosting. For the purpose of this paper, we assume three architectural models. The Fully-Shared model is based on a shared application instance and service components being shared by tenants

Table 1. Impact of the Architectural Choices on Service Variability

	Degree of Service Variability	**Cost of Service Variability**	**Benefit of Service Variability**
Service Engineering			
Static Binding	High Degree	High Cost	Large Tenant base
Dynamic Binding	Low Degree	Low Cost	Small to Medium Tenant base
Service Packaging			
Service Level Encapsulation (SLE)	Low Degree	Low Cost	Small to Medium Tenant base
Tenant Level Encapsulation (TLE)	High Degree	High Cost	Large Tenant base
Service Hosting			
Shared Instance (SI)	Low Degree	Low Cost	Small to Medium Tenant base
Dedicated Instance (DI)	High Degree	High Cost	Large Tenant base

during service execution. The Partially-Shared model is based on shared application instance but as compared to Fully-Shared model the software components in Partially-Shared model can be tenant-specific (TLE) or service-specific (SLE) or both. For example, the boundary components can be tenant-specific while the controller components are service-specific. The No-Shared model is based on each tenant having own, dedicated application instance and the service components are tenant-specific. The Fully-Shared model can only adopt dynamic binding techniques while the Partially-Shared and No-Shared models can adopt both static and dynamic techniques. Table 2 summarize these relationships.

Table 2. SaaS Architectural Models

Models	**Service Hosting**	**Service Packaging**	**Service Engineering**
Fully-Shared	Shared Instance	Service Level Encapsulation	Dynamic Binding
Partially-Shared	Shared Instance	Service or Tenant Level Encapsulation	Static or Dynamic Binding
No-Shared	Dedicated Instance	Tenant Level Encapsulation	Static or Dynamic Binding

The approach to determine the architectural model depends on the variations of service requirements, architectural choices and the cost/benefit analysis of the Service Providers. Fig. 2 illustrates this approach.

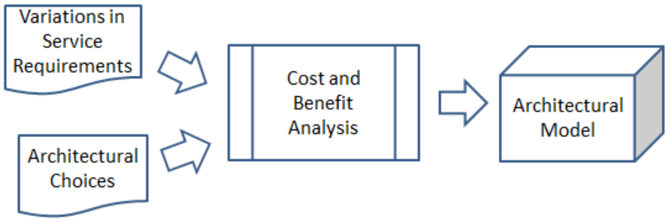


Fig. 2. Determination of the SaaS Architectural Model

4 Variability-Related Scenarios - Service Provider Perspective

Service Provider wants to employ SaaS solution that maximizes profitability of selling her application as a service. Therefore, the Service Provider needs an architectural model for a service that would lower the cost of the service offering and widen the tenants base.

1. *Lowest cost* - The Service Provider chooses the architectural model that incurs lowest cost to maximize profits. In particular for service variability, the Service Provider has to make decision to support service variability with the lowest cost. Based on Table 1 and 2, the Service Provider is likely to go for the Fully-Shared Model to minimize the cost. However, the lower degree of service variability imply that some tenants with high variations of service requirements cannot be met.
2. *Maximize Revenue* - The Service Provider needs to fulfill the tenant's expectations to widen the tenants base and increase revenue. The tenant expects their requirements to be fulfilled as if the service is single tenant. The higher degree service variability implies greater extent of the tenant's requirements that can be fulfilled. The Service Provider can go for No-Shared Model. The associated higher cost incurred by the Service Provider imply that the tenants have to be able to afford the higher fee.
3. *Tenant On-boarding (Low degree of service variability)* - The Service Provider wants to on board as many tenants as possible. However, the benefit of on boarding new tenants (increased revenue) should be weighed against the cost of adapting the service to possibly new requirements (i.e., the cost of service variability). In this example assuming there is an initially small number of tenants (e.g. 30 tenants) with low variation of requirements. In this case, the Service Provider chooses the Fully-Shared model to minimize the cost of service variability. If $\text{InfraSharedCost}_{(30)}$ is the shared infrastructure cost of supporting 30 tenants and $\text{CostDVTDesign}_{(30)}$ is the cost to implement the dynamic binding variability techniques, then the cost of offering an application as a service is:

$$\text{InfraSharedCost}_{(30)} + \text{CostDVTDesign}_{(30)}$$

4. *Tenant On-boarding (High degree of service variability)* - Assuming there are 50 more tenants (more diverse requirements) interested in the service with 30 existing tenants. The Service Provider can provide dedicated service instances for new tenants and applying both static and dynamic binding techniques to cater for variant requirements. In this case, the Service Provider needs to evaluate the overall cost of providing dedicated instances and implementing the static and dynamic binding techniques for a No-Shared model. If $\text{CostSDVTDesign}_{(50)}$ is the cost to re-design for static and dynamic binding and $\text{InfraDedicatedCost}_{(50)}$ is the dedicated infrastructure cost of supporting 50 tenants, then the cost of offering an application as a service is:

$$\text{InfraSharedCost}_{(30)} + \text{CostDVTDesign}_{(30)} + \text{InfraDedicatedCost}_{(50)} + \text{CostSDVTDesign}_{(50)}$$

   The Service Provider can also chooses to place the 50 tenants on with existing tenants in a Partially-Shared model. In this case, the Service Provider needs to evaluate the impact due to the higher variability of requirements. If CostDVTRedesign(50) is the cost to re-design for dynamic binding, then the cost equation for service variability is:

$$\text{InfraSharedCost}_{(30)} + \text{CostDVTDesign}_{(30)} + \text{InfraSharedCost}_{(50)} + \text{CostDVTRedesign}_{(50)}$$

   To on-board the 50 tenants, the Service Provider can make decisions based on the minimum cost of both choices.

$$\text{Min}\ (\text{InfraSharedCost}_{(50)} + \text{CostDVTRedesign}_{(50)}\ , \text{InfraDedicatedCost}_{(50)} + \text{CostSDVTDesign}_{(50)}\ )$$

   If the Service Provider is aware of the need to support up to 80 tenants(30 tenants with low variation of requirements and another 50 tenants having high degree variation of requirements), the service provider can alternatively plan to support the 80 tenants directly with No-Shared for all 80 tenants. If InfraDedicatedCost(80) is the cost for dedicated infrastructure to support 80 tenants and CostSDVTDesign(80) is the cost to implement the static and dynamic binding variability techniques, then the cost equation for service variability is:

$$\text{InfraDedicatedCost}_{(80)} + \text{CostSDVTDesign}_{(80)}$$

5. *Service Isolation* - To many organizations, security and privacy are still the top issues in adopting SaaS. The No-Shared model would be most suitable with software components and process instance being tenant-specific. Service Providers might want to propose Partially-Shared model (e.g. only the entity components are tenant-specific) for the group of tenants who are more price-sensitive.

5 Related Work

The profitability model for SaaS is an area that attracts much interest. The author of [1] propose an analytical SaaS cost model based on user's fit and exit costs. In [3], the author analysis the pricing strategies for SaaS and COTS. The authors of [4] attempts to maximize Service Provider's profit and tenant functional commonality for tenant onboarding in terms of contracts. In comparison, we evaluate profitability from both the costs and revenue perspectives with the architectural choices.

6 Conclusion

We addressed the problem of profitability of SaaS solutions in view of the revenues from selling the service, and the cost of SaaS computing infrastructure to offer a service to tenants. The first depends on the number of tenant who pay for the service. The latter is determined by the cost of computer resources utilization, and the cost of service engineering. We identified trade offs involved in Service Provider decisions regarding the choice of service variability (i.e., the ability to satisfy the diversity of tenants' requirements) and SaaS architecture for the service. With the growing number of tenants, the likelihood of facing more diverse tenants' requirements increases. We found that high service variability may call for more costly SaaS architectures (e.g., dedicated service instance as opposed to shared instance), and more costly techniques for service variability management (e.g., static binding as opposed to dynamic binding). We summarized the results of our analysis in tables that show influences among factors that determine profitability of the SaaS solution. We believe our results can help Service Providers make more informed decisions regarding service offering.
Our current study is qualitative. In future work, we will extend it with quantitative analysis and models that more precisely correlate SaaS costs and benefits, giving more accurate insights into profitability of SaaS from service variability perspective. We plan to decompose cost and benefit of SaaS solution into more detailed factors that will include the effort of migrating an existing application into a service and on-board new tenants.

References

1. Ma, Dan. "The business model of" software-as-a-service"." Services Computing, 2007. SCC 2007. IEEE International Conference on. IEEE, 2007.
2. Jarzabek, Stan, et al. "XVCL: XML-based variant configuration language." Software Engineering, 2003. Proceedings. 25th International Conference on. IEEE, 2003.
3. Ma, Dan, and Abraham Seidmann. "The pricing strategy analysis for the Software-as-a-service business model." Grid Economics and Business Models. Springer Berlin Heidelberg, 2008. 103-112.
4. Lei Ju, Bikram Sengupta, and Abhik Roychoudhury. "Tenant Onboarding in Evolving Multi-tenant SaaS." (2011).

Towards Configurable Data Collection for Sustainable Supply Chain Communication

Gregor Grambow, Nicolas Mundbrod, Vivian Steller and Manfred Reichert

Institute of Databases and Information Systems
Ulm University, Germany
{gregor.grambow,nicolas.mundbrod,vivian.steller,manfred.reichert}@uni-ulm.de
http://www.uni-ulm.de/dbis

Abstract. These days, companies in the automotive and electronics sector are forced by legal regulations and customer needs to collect a myriad of different indicators regarding sustainability of their products. However, in today's supply chains, these products are often the result of the collaboration of a large number of companies. Thus, these companies have to apply complex, cross-organizational, and potentially long-running data collection processes to gather their sustainability data. Comprising a great number of manual and automated tasks for different partners, these processes imply great variability. To support such complex data collection, we have designed a lightweight, automated approach for contextual process configuration.

Key words: Process Configuration, Business Process Variability, Data Collection, Sustainability, Supply Chain

1 Introduction

In todays' industry many products are the result of the collaboration of various companies working together in complex supply chains. Cross-organizational communication in such areas can be quite challenging due to the fact that different companies have different information systems, data formats, and approaches to such communication. These days, state authorities, customers and the public opinion demand sustainability compliance from companies, especially in the electronics and automotive sector. Therefore, companies have to report certain sustainability indicators as, e.g., their greenhouse gas (GHG) emissions or the amount of lead contained in their products. Such reports usually also involve data from suppliers of the reporting company. Therefore, companies launch a sustainability data collection process along their supply chain. This often involves also the suppliers of the suppliers and so on.

As sustainability data collection is a relatively new and complicated issue, service providers (e.g., for data validation or lab tests) are also involved in such data collection. A property that makes these data collection processes even more complex and problematic is the heterogeneity in the supply chain: companies use different information systems, data formats, and overall approaches to sustainability data collection. Many of them even do not have any information system or approach in place for this and answer with low quality data or not at all. Therefore, no federated system or database could be applied to cope with such problems and each request involves an often long-running, manual, and error-prone data collection process. The following simplified scenario illustrates issues with the data collection process in a small scale.

Scenario: Sustainability Data Collection
An automotive company wants to collect sustainability data relating to the quantity of lead contained in a specific part. This concerns two of the companies suppliers. One of them has an IHS in place, the other has no system and no dedicated responsible for sustainability. For the smaller company, a service provider is needed to validate the manually collected data to ensure that it complies with legal regulations. The IHS of the other company has its own data format that has to be explicitly converted to be useable. This simple scenario already shows how much complexity can be involved even in simple requests and gives an outlook on how this can look like in bigger scenarios involving hundreds or thousands of companies with different systems and properties.

In the SustainHub[1] project, we develop a centralized information exchange platform that supports sustainability data collection along the whole supply chain. We have already thoroughly investigated the properties of such data collection in the automotive and electronics sectors and published a paper about challenges and state-of-the-art regarding this topic [1]. With this paper, we propose an approach that enables an inter-organizational data collection process. The main point thereby is the capability of this process to automatically configure itself in alignment with the context of its concrete execution.

To guarantee the utility of our approach as well as its general applicability, we have started with collecting problems and requirements directly from the industry. This involved telephone interviews with representatives of 15 European companies from the automotive and electronics sectors, a survey with 124 valid responses from companies of these sectors, and continuous communication with a smaller focus group to gather more precise information. Among the most valuable information gathered there was a set of core challenges for such a system: as most coordination for sustainability data exchange between companies is done manually, it can be problematic to find the right companies, departments, and persons to get data from and also to determine, in which cases service providers must be involved (DCC1). Moreover, this is aggravated by the different systems and approaches different companies apply. Even if the right entity or person has been selected, it might still be difficult to access the data and to get it in a usable format (DCC2). Furthermore, the data requests rely on a myriad of contextual factors that are only manage implicitly (DCC3). Thus, a request is

[1] SustainHub (Project No.283130) is a collaborative project within the 7th Framework Programme of the European Commission (Topic ENV.2011.3.1.9-1, Eco-innovation).

not reusable because an arbitrary number of variants can exist for it (DCC4). A system aiming at supporting such data collection must explicitly manage and store the requests, their variants, all related context data, and also data about the different companies and support manual and automated data collection.

The remainder of this paper is organized as follows: Section 2 shows our general approach for data collection with processes. Section 3 extends this with additional features regarding context and variability. This is followed by a brief discussion of related work in Section 4 and the conclusion.

2 Data Collection Governed by Processes

The basic idea behind our approach for supporting data collection in complex environments is governing the whole procedure by explicitly specified processes. Furthermore, these processes are also automatically enacted by a PAIS (Process-Aware Information System) that is integrated into the SustainHub platform. That way, the process of data collection for a specific issue as a sustainability indicator can be explicitly specified by a process type while process instances derived from that type govern concrete data collections regarding that issue. Activities in such a process represent the manual and automatic tasks to be executed as part of the data collection by different companies. This approach already covers a number of the elicited requirements. It enables a centralized and consistent request handling (cf. DCC1) and also supports manual as well as automated data collection (cf. DCC2). One big advantage lies in the modularity of the realization as process. If a new external system shall be integrated, a new activity component can be developed while the overall data collection process does not need to be adapted. Finally, it also enables the explicit specification of the data collection process (cf. DCC4). By visual modeling the creation and maintenance of such processes is facilitated. However, the realization via processes can only be the basis for comprehensive and consistent data collection support. To be able to satisfy the requirements regarding contextual influences, various types of important data, and data request variants, we propose an extended process-based approach for data collection illustrated in Figure 1.

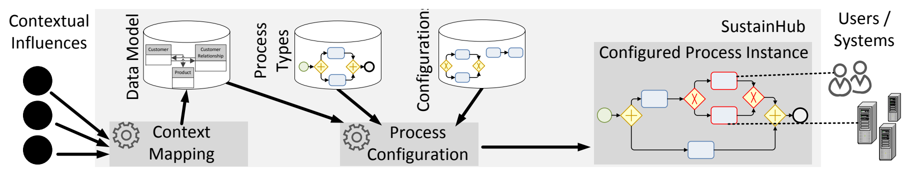


Fig. 1: SustainHub Configurable Data Collection Approach

To generate an awareness of contextual influences (e.g. the concrete approach to data collection in a company, cf. DCC3) and make them usable for the data collection process, we have defined an explicit context mapping approach (discussed in Section 3.1). This data is necessary for the central step of our approach, the automatic and context-aware process configuration (discussed in Section 3.2), where pre-defined process types and configuration options are used to automatically generate a process instance containing all necessary activities to match the properties of the current requests situation (cf. DCC4). As basis for this step, we have elaborated a data model where contextual influences are stored (cf. DCC3) alongside different kinds of content-related data. This data model integrates process-related data with customer-related data as well as contextual information. We will now briefly introduce the different kinds of incorporated data by different sections of our data model. At first, such a system must manage data about its customers. Therefore, a customer data section comprises data about the companies, like organizational units or products. Another basic component of industrial production that is important for many topics as sustainability are substances and (sustainability) indicators. As these are not specific for one company, they are integrated as part of a master data section. In addition, the data concretely exchanged between the companies is represented within a separate section (exchange data). To support this data exchange, the system must manage certain data relating to the exchange itself (cf. DCC1): For whom is the data accessible? What are the properties of the requests and responses? Such data is captured in a runtime data section in the data model. Finally, to be able to consistently manage the data request process, concepts for the process and its variants as well as for the contextual meta data influencing the process have been integrated with the other data. More detailed descriptions of these concepts and their utilization will follow in the succeeding sections.

3 Variability Aspects of Data Collection

This section deals with the necessary areas for automated process configuration: The mapping of contextual influences into the system to be used for configuration and the modeling of the latter.

3.1 Context Mapping

As stated in the introduction, a request regarding the same topic (in this case, a sustainability indicator) can have multiple variants that are influenced by a myriad of possible contextual factors (e.g. the number of involved parties or the data formats they use). Hence, if one seeks to implement any kind of automated variant management, a consistent manageable way of dealing with these factors becomes crucial. However, the decisions on how to apply process configuration and variant management often cannot be mapped directly to certain facts existing in the environment of a system. Moreover, situations can occur, in which different contextual factors will lead to the same decision(s) according to variant management. For example, a company could integrate a special four-eyes-principle approval process for the release of data due to different reasons like if the data is for a specific customer group or if the data relates to a specific law or regulation. Nevertheless, it would be cumbersome to enable automatic variant management by creating a huge number of rules for each and every possible contextual factor. Therefore, in the following, we propose a more generic way of mapping for making contextual factors useable for decisions regarding the data collection process.

In our approach, contextual factors are abstracted by introducing two separate concepts in a lightweight and easily configurable way: The *Context Factor* captures all different possible contextual facts existing in the systems' environment. Opposed to this, the *Process Parameter* is used to model a stable set of parameters directly relevant to the process of data collection. Both concepts are connected by simple logical rules as illustrated on the left side of Figure 2. In this example, a simple mapping is shown. If a contact person is configured for a company (CF1), the parameter 'Manual Data Collection' will be derived. If the company is connected via a tool connector (CF2), automatic data collection will be applied (P3). If the company misses a certain certification (CF3), an additional validation is needed (P2).

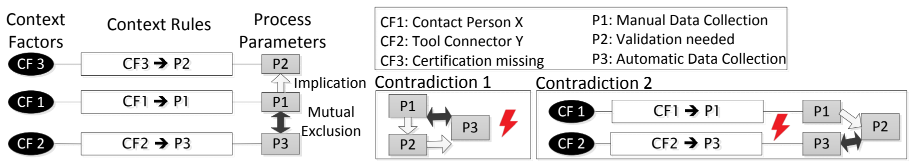


Fig. 2: Context Mapping

When exchanging data between companies, various situations might occur, in which different decisions regarding the process might have implications on each other. For example, it would make no sense to collect data both automatically and manually for the same indicator at the same time. To express that we have also included the two simple constraints 'implication' and 'mutual exclusion' for the parameters. For an example, we refer to Figure 2, where, for example, manual and automatic data collection are mutually exclusive.

Although we have put emphasis on keeping the applied rules and constraints simple and maintainable, there can still exist situations, in which these lead to contradictions. One case (Contradiction 1 in Figure 2) involves a contradiction only created by the constraints, where one activity requires and permits the occurrence of another activity at the same time. A second case (Contradiction 2 in Figure 2) occurs when combining certain rules with certain constraints, in which a contradicting set of parameters is produced. To avoid such situations, we have integrated a set of simple correctness checks for constraints and rules.

3.2 Process Configuration

In this section, we will introduce our approach for process configuration. Therefore, we not only considered the aformentioned challenges, we also wanted to keep the approach as easy and lightweight as possible to enable users of SustainHub to configure and manage the approach. Furthermore, our findings included data about the actual activities of data collection and their relation to contextual data. Data collection often contains a set of basic activities that are part of each data collection process. Other activities appear mutually exclusive, e.g. manual or automatic data collection, and no standard activity can be determined here. In most cases, one or more context factors impose the application of a set of additional coherent activities rather than one single activity.

In the light of these facts, we have opted for the following approach for automatic process configuration: For one case (e.g. a sustainability indicator) a process family is created. The latter contains a *Base Process* with all basic activities for that case. Additional activities that are added to this Base Process are encapsulated in *Process Fragments*. These are automatically added to the process on account of the parameters of the current situation that is represented in the system by the already introduced *Process Parameters* and *Context Factors*. Thus, we only rely on one single change pattern to the processes, an insert operation. This operation has already been described in literature, for its formal semantics, see [2]. Thus our approach avoids problems with other operations as described by other approaches like Provop [3]. Figure 3 shows a simple example of a Base Process that has been configured with Process Fragments (configured areas are marked red). For simplicity, this example uses a subset of the activities of the scenario from the introduction.

To keep the approach lightweight and simple, we decided to model both the Base Process and the fragments in a PAIS (Process-Aware Information System) that will be integrated into our approach. Thus, we can rely on the abilities of the PAIS for modeling and enacting the processes and also for checking their correctness.

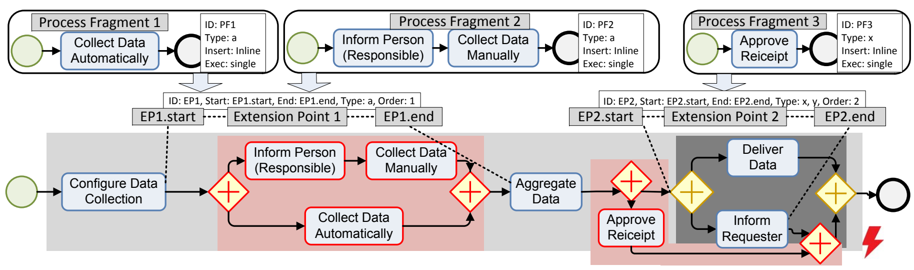


Fig. 3: Process Fragments

To enable the system to automatically extend the base process at the right points with the chosen fragments, we have added the concept of the *Extension*

Point (EP). Both the latter and the fragments have parameters, the system can match to find the right EP for a fragment (see Figure 3 for two example EPs and three fragments with matching parameters). Regarding the connection of the EPs to the Base Processes, we have also evaluated multiple options as, e.g., connecting them directly to activities. Most of such options introduce limitations to the approach or impose a fair amount of additional complexity (cf. [3] for a more detailed discussion). For these reasons we have selected an approach involving two so-called *connection points* of an EP with a Base Process. These points are connected with nodes in the process as shown in Figure 3. Taking the nodes as connection points allows us to reference the nodes' Id for the connection point because this Id is stable and would only change in case of more complicated configuration actions (cf. [3]). If the Base Process contains nodes between the connection points of one EP, an insertion would be applied in parallel to these (cf. EP2 in Figure 3), otherwise sequentially (cf. EP1). Furthermore, if more than one fragment should be inserted at one EP, they will be inserted in parallel to each other (cf. EP1 and fragments 1 and 2 in Figure 3).

By relying on the capabilities of the PAIS we have kept the number of additional correctness checks small. However, the connection points are not checked by the PAIS and could impose erroneous configurations. To keep correctness checks on them simple we rely on two things: The relation of two connection points of one EP and block-structured processes [4]. The first fact spares us from having to check all mutual connections of all connection points as two always belong together. The second implies certain guarantees regarding the structure of the processes. So we only have to check a small set of cases, as e.g., the erroneous definition of EP2 in Figure 3 that would cause a violation to the block structure as shown in the figure.

4 Related Work

Regarding the topic of process configuration, various approaches exist. Most of them focus on the modeling of process configuration. One example is C-EPC [5] that enables behavior-based configurations by integrating configurable elements into a process model. Another approach with the same focus is ADOM [6]. It allows for the specification of constraints and guidelines on a process model to support variability modeling. For all of these approaches two main shortcomings apply: First, they strongly focus on the modeling and neglect execution. Second, configuration must be manually applied by a human, which can be complicated and time-consuming. The approach most closely related to ours is probably Provop [3]. It allows storing a base process and pre-configured configurations to it. Compared to our approach Provop is more fine-grained, complicated, and heavyweight whereas our approach utilizes a set of simplifications that enable a far more lightweight approach. For further reading on the configuration topic, see [7] for an overview of configuration approaches and our predecessor paper for SustainHub [1].

5 Conclusion

In this paper, we have shown a lightweight approach to automatic and contextual process configuration required in complex domains. We have investigated concrete issues in an example domain relating to sustainability data collection in supply chains. With our approach, we have centralized the data and process management uniting many different factors in one data model and supporting the whole data collection procedure by processes executed in a PAIS. Moreover, we have enabled this approach to apply automated process configurations conforming to different situations by applying a simple model allowing for mapping contextual factors to parameters for the configuration. In future work, we plan to evaluate our work with our industrial partners and to extend our approach to cover further aspects regarding runtime variability, automated monitoring, and automated data quality management.

Acknowledgement

The project SustainHub (Project No.283130) is sponsored by the EU in the 7th Framework Programme of the European Commission (Topic ENV.2011.3.1.9-1, Eco-innovation).

References

1. Grambow, G., Mundbrod, N., Steller, V., Reichert, M.: Challenges of applying adaptive processes to enable variability in sustainability data collection. In: 3rd Int'l Symposium on Data-Driven Process Discovery and Analysis. (2013) 74–88
2. Rinderle-Ma, S., Reichert, M., Weber, B.: On the formal semantics of change patterns in process-aware information systems. In: Proc. 27th Int'l Conference on Conceptual Modeling (ER'08). Number 5231 in LNCS, Springer (October 2008) 279–293
3. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. In: Int'l Handbook on Business Process Management I. Springer (2010) 237–255
4. Mendling, J., Reijers, H.A., van der Aalst, W.M.: Seven process modeling guidelines (7pmg). Information and Software Technology **52**(2) (2010) 127–136
5. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. Information Systems **32**(1) (2005) 1–23
6. Reinhartz-Berger, I., Soffer, P., Sturm, A.: Extending the adaptability of reference models. IEEE Trans on Syst, Man, and Cyber, Part A **40**(5) (2010) 1045–1056
7. Torres, V., Zugal, S., Weber, B., Reichert, M., Ayora, C., Pelechano, V.: A qualitative comparison of approaches supporting business process variability. In: 3rd Int'l Workshop on Reuse in Business Process Management (rBPM 2012). BPM'12 Workshops. LNBIP, Springer (September 2012)

A Meta-Model for Process Map Design

Monika Malinova[1], Henrik Leopold[1], and Jan Mendling[1]

WU Vienna, Welthandelsplatz 1, A-1020 Vienna, Austria
monika.malinova|henrik.leopold|jan.mendling@wu.ac.at

Abstract. Process maps provide a holistic view of all processes of an organization by visualizing their essential relationships. The design of a process map is of central importance as many organizations create them at the start of a business process management (BPM) initiative to serve as a framework. Despite this importance, the design of process maps is still more art than science, essentially because there is no standardized modeling language available for process map design. In this paper, we address the research question of which concepts are currently used in process maps and how they are related to each other. To this end, we investigate 67 process maps. Our contribution is a meta-model for process map design which is grounded in actual usage. Furthermore, we discuss the importance of different concepts for process map design.

Keywords: process architecture, process map, meta-model

1 Introduction

Process maps are a key concept for providing an overview of a company's business processes [1]. They visualize the main relationships between processes and facilitate a basic understanding of how the company operates. The importance of process maps is illustrated by the growing extent of process modeling initiatives in practice. Often companies maintain process model collections with thousands of process models [2]. Typically, creating a process map is the first task when introducing Business Process Management (BPM) into an organization as it provides an abstract view of all processes [3]. Thus, the process map is often used as a guide for conducting the subsequent steps of the BPM lifecycle [1].

While there is some initial research on abstraction and categorization of model collections [4, 5, 6], there is notable insecurity on how to capture such process-related information on the most abstract level of a process architecture [7]. In practice, process maps are used for that purpose, however, without a standard modeling language being available. On the other hand, it has been found that an incomplete and incorrect process map design could have negative effects on the BPM success altogether [1]. The major challenge in this context is the specification of a language for process map design that integrates insights from actual usage in practice.

In this paper, we make the first step towards addressing this research gap by conducting an explorative study. We investigate the included concepts of existing 67 process maps. As a result we present a process map meta-model which describes the current state of process map design i.e. all concepts we observed and the relations between them. We also investigate patterns of usage of these concepts. In this way, we aim to provide a foundation for the standardization of a language for process map design.

The rest of the paper is structured as follows. Section 2 gives a brief overview of BPM and process maps. Section 3 introduces the process maps we used along with the methods we applied to derive our findings. Section 4 presents the results of our study. Section 5 points to some implications for research and practice, while Section 6 concludes the paper.

2 Background

In this section, we give insights into the organization of business processes within an organization and we present the current state of the art of process maps.

2.1 Organizing business processes

A business process consists of activities that when executed transform inputs into outputs [8]. Typically, a sequence of such processes is performed in order to create a value for the customer [3]. However, processes may differ in their importance for value creation. Thus, they are commonly categorized based on the degree of their proximity to customers. To manage interrelations between the processes and to systematically document how the firm operates as a whole, organizations often adopt the BPM approach and start modeling their processes in form of process models. A process model visualizes the process steps by providing a diagrammatic representation of a singular process. As a result of such modeling initiatives, organizations often end up with a large collection of process models. A process architecture helps to store all detailed process models and the relations between them in a systematic manner [7]. A process map is typically used as the top level in a process architecture. It visualizes all processes and their relationships in a compact way [1].

2.2 Process maps

We can trace back the concept of process maps to the early 1980s when Porter introduced the value-chain model. The value chain provides a process view of an organization and represents it as a set of core activities a firm performs in order to create value for the customer [9]. Scheer adopts the concept of a value-added chain [10]. He introduces a diagram that represents those processes that create value for the company. These processes are shown in a sequence, and each can be hierarchically decomposed into subprocesses that a super-ordinate process needs in order to be executed [10]. SIPOC is another frequently used tool[11, 12]. It stands for supplier, inputs, process, outputs, and customers and is used as a guide for analyzing these five aspects with main focus on the customer [12].

Examples of process maps can also be found in literature [3, 12, 13, 14]. Most are based upon the value-chain concept and provide means of identifying process categories and the role each type of process plays for the company. Generally, those processes that directly create value for the customer and generate revenue are called *core processes* [1, 14]. In a process map, these processes are usually related to each other in a sequential manner and represented as end-to-end processes [15]. An end-to-end process is commonly a cross-functional process, i.e. a process that goes through more than one organizational unit [15]. In addition to core processes, there are also support and management processes. *Support processes* provide resources to the core processes, such as human resource management, information technology, etc. [14]. Whereas, *management processes* ensure that the execution of the core processes is aligned with the company's strategy [14]. In addition to process categories, a process map also depicts relationships between the processes. The notion of input/output can as well be very often observed in process maps.

However, whereas there are well-defined languages for modeling singular processes (e.g. BPMN, EPC), such a language for process map design is missing, which can be inferred from the high heterogeneity of process map designs we see in practice. To our knowledge, no research has been conducted on the extent to which process map elements serve all the representational needs of process map designers. In this study we aim to consolidate the current practice of process map design in order to provide a foundation for developing a language that helps practitioners to design process maps in a standardized manner.

3 Research Design

In order to understand the current practice of process map design, we analyze process maps from literature and practice. We want to (1) elicit meaning and develop knowledge into the concepts used within a single process map and (2) find patterns of the combined usage of concepts.

3.1 Methods

To address the first point, we gather process maps and analyze each of them for the concepts being used and any means by which the identified concepts relate to each other. As a result, we generate a process map meta-model which encapsulates all concepts and relations we observed. We use UML (Unified Modeling Language) as a language to design the meta-model. To address the second point, we adopt the approach of [16]. We create an Excel spread sheet and record each concept and relation between the concepts. We encode the usage of each with 1 or 0. We apply hierarchical clustering on the data to identify concepts that frequently or rarely occur together in a specific combination within a single map.

3.2 Data collection

We analyze process maps from both practice and literature. We found 21 process maps in BPM books [13, 14, 17, 18, 19, 20]. In addition, from interviews we conducted with companies, 13 of them provided us with a print out of their process map. Also, we used 5 process maps that were part of published case studies [21]. In order to make sure we cover all concepts used in existing process maps, we searched for additional process maps using an Internet search engine. For this we used two key words, namely "process map" and "process landscape". Altogether, we use 67 process maps in the analysis.

4 Findings

In this section we present the meta-model, explain the included concepts and show the results of the hierarchical cluster analysis.

4.1 Process map meta-model

The process map meta-model in Figure 1 depicts all unique concepts we found in the 67 process maps. The key component of process maps is a business process. A *process* is triggered by an input from the supplier and is usually clustered in a category with other processes that serve a similar purpose. A process could also belong to a phase depending on the time of execution. They are conducted by actors, could eventually use a resource during their execution and can be related to other processes in order to produce an output for the customer.

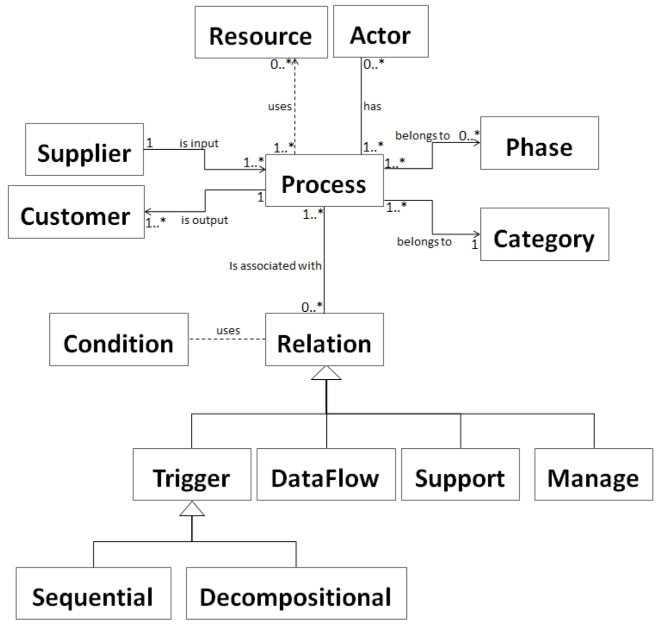
A *supplier* is a party that provides inputs that triggers the execution of an end-to-end process. A *customer* is the one who receives outputs resulting from the execution of a process. A *resource* is a source of supply or aid that can be drawn upon when needed by any process or an instance of a process (e.g. resource *water* is required during the production of energy). If necessary, one process uses one or more resources throughout its execution. However, a process does not necessarily need to use a resource in order to produce an output. One process can have zero or more *actors* that are responsible for its performance.

A *category* is a group of processes that have a particular role within one company. One process can belong to only one category. Processes that are clustered in one category serve a similar purpose. A *phase* is a temporal category that contains a subset of processes coming from one or more process categories. It is temporal because a certain number of processes need to be performed in order for an intermediate outcome to be produced. This intermediate outcome could be used as a trigger for the processes that belong to the next phase.

The *condition* constraints or guards the relation that is used between two or more processes (e.g. if process C can only start after processes A and B, than the condition will rule-out all those relations that do not capture this behavior).

One process can be related to other processes through one or more *relation* types. There are four main process relations: trigger, data flow, support, and manage.

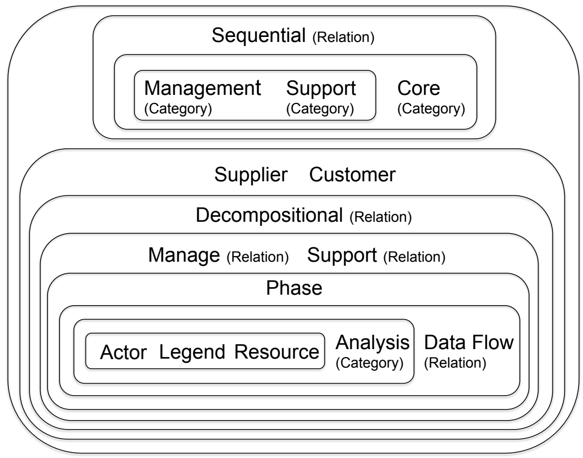
Fig. 1. Process map meta-model



The *trigger* relations could be used between processes that belong to the same or to different process categories. There are two types of trigger relations. (1) A *sequential* trigger is a control-flow relation used between processes to indicate order of performance. Hence, only when the first process finishes the second process can start with execution. Alternative variations of process order are also possible, such as when one process is finished with execution it could trigger more than one process, accordingly processes could also be executed in parallel and one process could be triggered several times in a row until the desired outcome is produced. (2) A *decompositional* trigger relates a core process to its subprocesses. If a process is hierarchically decomposed it has a number of subprocesses that need to be executed in order for the process to finish.

Data flow can be depicted between processes that belong to the same or different categories. This relation, when used, does not necessarily trigger another process. Instead, it only passes information from one process to another without interrupting its performance. A *support* relation is used only between the processes from the core and support process categories. The direction of support goes from the support to the core processes. Support processes serve any immediate need by all of the core processes. Likewise, a *manage* relation is used only between processes that belong to the core and management process categories. The management processes manage the core processes by taking care that the process is performed according to defined rules.

Fig. 2. Hierarchical clusters of process map concepts



4.2 Use of process map concepts

Figure 2 illustrates the result of a hierarchical cluster analysis. The figure gives us insights into the design of process maps. In particular, it shows us which concepts are most frequently used together in a combination within one process map. We observe that there is apparently a connection between the three process categories and the sequential relation. Namely, this is the most frequent combination of concepts that usually appears in a single process map as seen in the upper part of Figure 2.

Concerning the *categories*, we learn that the management and the support categories typically occur together. This indicates that organizations either focus only on core processes, or in case they include other categories, than they show both the management and the support category.

With the *relations* we can also identify a clear pattern. It appears that there are different types of process maps. While some maps provide many details including several types of relations, other maps aim at providing a broad overview and omit relations other than the sequential trigger. As for the *supplier* and the *customer*, we observe that both are part of one cluster. This indicates that these concepts are very likely to occur together. So, organizations tend to either show both or none of them.

Concerning the rest of the concepts, we observe two aspects. First, actors, resources, and legends are part of one cluster. Hence, their occurrence is positively correlated, which means that either a process map provides all or it includes none. The second point relates to the co-occurrence of actors, resources, and data flow. As indicated by the clusters, there is a tendency for these concepts to occur together. This illustrates that maps containing actors and resources are also more likely to show how these resources are used during the entire process flow, until the point an output is produced.

The hierarchical clustering points to the fact that process maps might be used with differing intentions. While some maps provide extensive detail such as actors, resources, and triggers, other are rather inclined to provide an abstract picture. The latter category tends to omit concepts like data flow relations and other details such as actors and resources.

5 Implications

The findings from this paper have several implications for research and practice. In relation to implications for practice, we emphasize the importance of process map design completeness. We argue that a well-designed process map should be able to elicit basic understanding of the company's operations. For this, the designer should use all those concepts that will enable such understanding, rather than only represent the company's processes in three categories without using any additional elements. Also, taking into account that a process map design is considered as a strategic step, its design could strongly influence the subsequent detailed process modeling [1].

In terms of implications for research, the analysis we present provides a basis for consolidation of concepts and represents a step towards a standardized language for process map design. Thus, this paper sets a starting point for their design by summarizing all used concepts in currently existing process maps. This particularly assists in establishing a body of knowledge on current process map design.

6 Conclusion

In this paper, we investigated the concepts used in 67 process maps from literature and practice. Based on this analysis, we derived a process map meta-model that covers all concepts these process maps use, as well as the relations between them. We applied a hierarchical clustering method and showed the most frequent combinations of concepts used within a single process map.

We found that the core process category is used in all process maps, while those maps that include a support category also have a tendency to include the management process category. The sequential relation is frequently used by most maps to relate processes. In addition, our findings showed that those process maps that include additional information beyond process categories and relations, such as actors, are also likely to include extra concepts, such as resources, and show how all this information is presented throughout the process execution with the use of a data flow relation.

References

1. Malinova, M., Mendling, J.: The effect of process map design quality on process management success. In: Proceedings of the 21th European conference on Information Systems. (2013)

2. Rosemann, M.: Potential Pitfalls of Process Modeling: Part A. Business Process Management Journal **12**(2) (2006) 249–254
3. Dumas, M., Rosa, M., Mendling, J., Reijers, H.: Fundamentals of Business Process Management. Springer (2013)
4. Leopold, H., Mendling, J., Reijers, H.A., Rosa, M.L.: Simplifying process model abstraction: Techniques for generating model names. Inf. Syst. **39** (2014) 134–151
5. Malinova, M., Dijkman, R., Mendling, J.: Automatic extraction of process categories from process model collections
6. Smirnov, S., Reijers, H.A., Weske, M., Nugteren, T.: Business process model abstraction: a definition, catalog, and survey. Distributed and Parallel Databases **30**(1) (2012) 63–99
7. Malinova, M., Leopold, H., Mendling, J.: An empirical investigation on the design of process architectures. In: Wirtschaftsinformatik. (2013)
8. Kiraka, R.N., Manning, K.: Managing organisations through a process-based perspective: its challenges and benefits. Knowledge and Process Management **12**(4) (2005) 288–298
9. Porter, M.E.: Competitive advantage: Creating and sustaining superior performance. SimonandSchuster. com (2008)
10. Scheer, A.: Aris: Business Process Modeling. Springer (2000)
11. Eckes, G.: The Six Sigma revolution: How General Electric and others turned process into profits. Wiley. com (2002)
12. Harmon, P.: Business process change: a guide for business managers and BPM and six sigma professionals. Morgan Kaufmann (2010)
13. Fischermanns, G.: Praxishandbuch Prozessmanagement. Schmidt (2006)
14. Mahal, A.: How Work Gets Done: Business Process Management, Basics and Beyond. Technics Publications, LLC (2010)
15. Maddern, H., Smart, P.A., Maull, R.S., Childe, S.: End-to-end process management: implications for theory and practice. Production Planning & Control (ahead-of-print) (2013) 1–19
16. zur Muehlen, M., Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: Proceedings of the 20th international conference on Advanced Information Systems Engineering. CAiSE '08, Berlin, Heidelberg, Springer-Verlag (2008) 465–479
17. Jeston, J., Nelis, J.: Business process management: practical guidelines to successful implementations. Routledge (2008)
18. Becker, J., Kugeler, M., Rosemann, M.: Process management: a guide for the design of business processes: with 83 figures and 34 tables. Springer Verlag (2003)
19. Weske, M.: Business Process Management: Concepts, Languages, Architectures. 2nd edn. Springer (2012)
20. Franz, P., Kirchmer, M.: Value-driven Business Process Management: The Value-switch for Lasting Competitive Advantage. McGraw-Hill (2012)
21. Kern, E.M.: Prozessmanagement individuell umgesetzt: Erfolgsbeispiele aus 15 privatwirtschaftlichen und öffentlichen Organisationen. Springer (2012)

Decision Point Analysis of Time Series Data in Process-Aware Information Systems*

Reinhold Dunkl, Stefanie Rinderle-Ma,
Wilfried Grossmann, Karl Anton Fröschl

University of Vienna, Faculty of Computer Science
{reinhold.dunkl, stefanie.rinderle-ma, wilfried.grossmann,
karl-anton.froeschl}@univie.ac.at

Abstract. The majority of process mining techniques focuses on control flow. Decision Point Analysis (DPA) exploits additional data attachments within log files to determine attributes decisive for branching of process paths within discovered process models. DPA considers only single attribute values. However, in many applications, the process environment provides additional data in form of consecutive measurement values such as blood pressure or container temperature. We introduce the $DPA^{TimeSeries}$ method as an iterative process for exploiting time series data by combining process mining and data mining techniques. The method also offers different approaches for incorporating time series data into log files in order to enable existing process mining techniques to be applied. Finally, we provide the simulation environment $DPA^{TimeSeries}_{Sim}$ to produce log files and time series data. The $DPA^{TimeSeries}$ method is evaluated based on an application scenario from the logistics domain.

Keywords: Process Mining, Decision Mining, Data Mining, Time Series Data

1 Introduction

Process mining aims at discovery and analysis of process models based on event logs. So far, process mining methodology emphasized the control flow, that is, restricting analysis to time-stamped event data (so-called log files) gathered from, or produced by, executed process instances. An extension towards the branching logic of processes is provided by Decision Point Analysis (DPA) [1]. DPA is based on enriching log file entries with additional information about process environments or other process-relevant data and aims at deriving decision rules at alternative branchings in process models. In a first step, the underlying process model is discovered. If the resulting process model contains decision points, the corresponding decision rules are analyzed using decision trees (data mining).

Fig. 1 depicts a container transportation example [2], where some temperature-sensitive cargo is transported and cargo temperature is measured repeatedly. On

* The work presented in this paper has been partly conducted within the EBMC^2 project funded by the University of Vienna and the Medical University of Vienna.

the left, the application of DPA [1] is illustrated: depending on the temperature value for each transport monitored, DPA concludes that for a temperature over 37, the vehicle has to return to its home base. Otherwise, it unloads the goods at the destination. As this example shows i) DPA takes into consideration single-valued attributes; ii) DPA is able to derive decision rules of type "x OP value" where x is the decision variable and OP is a comparison operator; iii) DPA relies on values that are stored within the event log of a process.

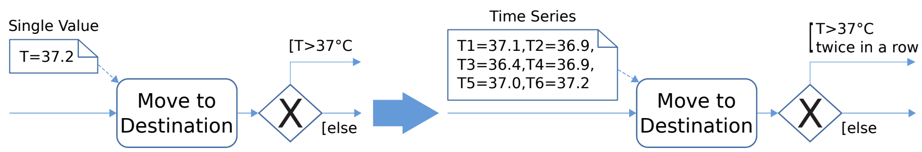


Fig. 1. Process Applications with Time Series Data

As the above characteristics show, DPA cannot adequately deal with real-world scenarios in which *time series data* are collected, e.g., in health care or container transportation. Based on time series data, *more complex* decision rules are conceivable, for example, "temperature exceeds a certain threshold for a time frame" (cf. right side in Fig. 1).

Hence, it would be desirable to process and analyze time series data by an extension of DPA. In this paper, we will present such an extension by means of method $DPA^{TimeSeries}$ that enables (a) a joint consideration of event log data and time series data, (b) iterative application of process and data/visual mining techniques, and (c) derivation of complex decision rules.

To do so, we distinguish two pertinent perspectives of this enhanced approach to process mining, viz. a method and a data perspective (Section 2). The ensuing process mining method is evaluated based on a real-world example of process analysis (Section 3). After reflecting our contribution against the state of the art in process and decision mining (Section 4), some concluding remarks (Section 5) finish this presentation.

2 Method and Data Perspective

The $DPA^{TimeSeries}$ method is illustrated in Fig. 2. As a first step it has to be decided how time series data is considered in connection with the event log data. For offering data structures within or outside the event logs that enable the application of $DPA^{TimeSeries}$, we identify the following options (cf. Fig. 2):

1. *Separation of Data:* We can prepare an analytical data set consisting of recurring measurements with sufficient temporally information to enable a matching with event data and provide this data separated from the log files.

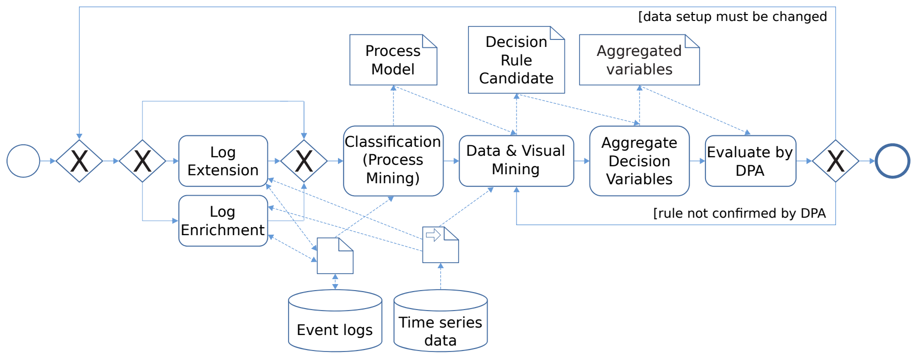


Fig. 2. $DPA^{TimeSeries}$ method (in BPMN notation)

2. *Log Enrichment:* This analytical data set can also be incorporated into the log by adding an attribute to the corresponding event within the log (e.g., a XES extension that allows such recurring measurement data structures).
3. *Log Extension:* Another approach is to dissemble the recurring measurement data and interlacing it into the log file as recurring events with single-valued attributes.

In the following, we discuss the pros and cons of these different options.

Separation of data does not modify the original event log data and therefore contributes to the maintenance of both data sets, an advantage if the event log data is used by other applications as well. The obvious disadvantage is that the connection between the event log data and the time series data is not explicitly stored and every analysis tool has to load and match the data by itself. Log enrichment and extension leads to an explication of this relation with the disadvantage of an additional preprocessing step to do so. Log enrichment does not change the number or kind of log entries as log extension does. Thus, process mining algorithms are not effected and, in turn, the resulting process models do not become more complex. Hence, the integration is in principle easier than for log extension. Log extension practically pushes the time series data into the event log what might be intended depending on the application and can therefore be an advantage as well as a disadvantage. This approach sure changes the log effectively but makes format extensions and extra files dispensable.

In summary, the choice of the approach is strongly dependent on the application. The case study presented in Sect. 3 features all three approaches.

As second step in the $DPA^{TimeSeries}$ method, process mining is used for *classifying* process execution paths along decisions made at runtime reflected by decision points in the resulting process models. In a third step we use data mining techniques such as CART, AdaBoost, Support Vector Machines as well as exploratory data mining including visual mining to explain the classification, i.e., derive the underlying decision rule. This more experimental mode of analysis, utilizing continuously improved understanding of (perhaps not yet) available process and environment data is more appropriate at this stage of the method than a mechanical brute-force exploration.

Candidates for decision rules are transformed into aggregated variables in a fourth step. These variables can then be used to employ DPA [1] to evaluate the decision rule candidate as the last fifth step. Depending on the result, the inspection by both data and visual mining techniques has to be repeated. It is also possible that the way the time series data was reflected inside or outside the logs has to be modified.

Process Mining uses event logs that consist of a minimal data set of case ids, activity names and timestamps. It is also possible to store data values that were produced during process execution, e.g., the age of a patient. These single-valued attributes are exploited by, for example, DPA. However, existing event log formats do not offer straightforward means to store time series data.

3 Evaluation

We start our evaluation by simulating the process of a container transport example adapted from [2] with an exact knowledge of the (complex) decision rules. After that we analyzed the log by integrating recurring measurement data using the proposed $DPA^{TimeSeries}$ method. In each iteration of the $DPA^{TimeSeries}$ we can compare the found decision rules with the original ones.

For the generation of process log data as well as time series data produced by recurring events within the iterations we implemented the simulation environment $DPA^{TimeSeries}_{Sim}$. Using a programming language like Java instead of a model interpreting tool like CPN-Tools [3] for simulation purpose gives us the flexibility to implement more complex rules. The time series data was integrated into the event data in various ways and exported in the log file format MXML to be used in ProM 5.2. Additionally, the time series data were exported in a simple CSV file to be used for data mining independent of the ProM framework. We used various mining algorithms from the ProM 5.2 framework to mine the models we used as a basis for DPA.

The basic idea of the container transport example is that some temperature-sensitive cargo is moved, implying that there is some temperature threshold not to be exceeded during the handling; otherwise, if this threshold is violated for a certain duration, the carriage is interrupted, and the transporting vehicle returns to its home base. Apparently, the decision whether to continue or interrupt the carriage depends on the monitored cargo temperature, measured by some sensor, for instance every 10 minutes as long as the vehicle moves towards its destination.

We now start the first iteration of the $DPA^{TimeSeries}$ method – based on this description of the process – with a simple simulation to obtain a first data set. 100 process instances are generated synthetically with up to 12 temperature measurements, such that in 30% of the cases the preset temperature threshold of 38°C is exceeded at least twice consecutively – in which case the carriage has to interrupt – whereas in 20% of the cases the threshold value is exceeded once at a time only, and in the remaining 50% of the process instances the threshold value is not overshot at all; that is, in 70% of the process instances the haulage continues until the destination is reached.

Using this data and the alpha algorithm of ProM 5.2[1] we develop the model shown in Fig. 3 (first model). We define a new analysis path for a better understanding of the decision of interrupting the carriage or not and identify that the temperature monitoring may be a useful candidate for a decision mining activity. Using the monitoring data as additional attribute and the approach of Log Enrichment (cf. Sect. 2) we attach the sequence of temperature observations to an "On the Way" event, after which the activities "Unload at Destination" (successful carriage) or "Return to Parking Lot" (interruption) commence.

A straight-forward application of the ProM 5.2 plug-in for DPA uses decision trees to identify attribute-value clauses underlying the branching of the process as shown in Fig. 3 (first model, shaded area). In order to apply this automatic procedure a data-preparing step is in place as the added time series in one attribute cannot be interpreted by the DPA. As the procedure would always refer to the latest (temperature) measurement available, an attribute indicating the most recent temperature observation at the time of branching was defined. In the following new iteration of the $DPA^{TimeSeries}$ method, DPA is able to classify all of the "Return to Parking Lot" instances correctly. However, due to the fact that the event of overshot temperature occurs at differing times DPA cannot infer a correct decision rule, because, for 20% of the instances taking the "Unload at Destination" branch, the overshot temperature condition is also met.

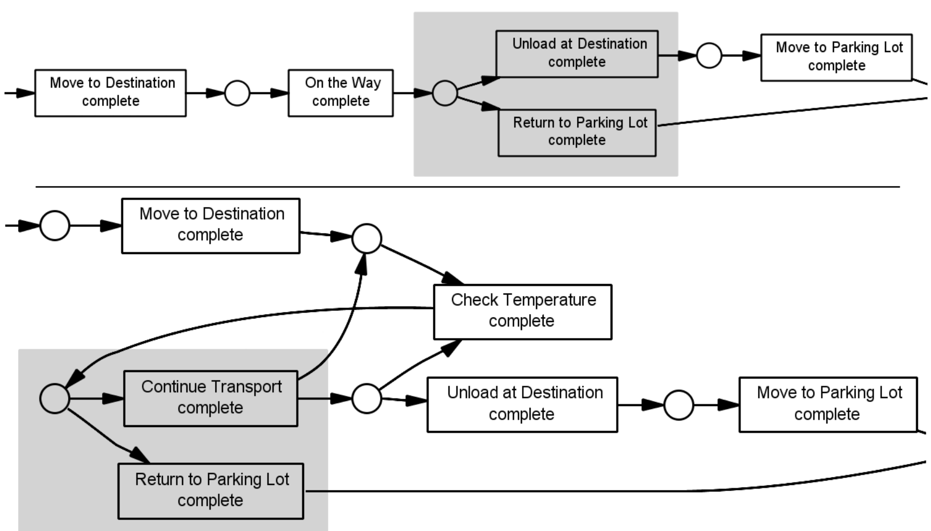


Fig. 3. Derived models, based on log enrichment and log extension (using ProM 5.2)

[1] `http://promtools.org/prom5/`

When starting a new iteration of the $DPA^{TimeSeries}$ method by adding all 12 (possible) measurements as individual process environment attributes (thus, however, losing the temporal *ordering* of the temperature information), DPA generates a fairly complex classification of cases able to classify 99 of the instances correctly anyway, but the tree is over-fitted to the data and fails to detect the proper decision criterion. The decision tree does not take into account the temporal ordering of the observations and is only applicable if all measurements are available. But even in that case it has only poor predictive power.

For the next iteration of the $DPA^{TimeSeries}$ method, we replaced the singular "On the Way" event of the process model, with all the temperature data attached, by a couple of recurrent activities, viz. "Check Transport" and "Continue Transport". This time, "Check Transport" events carry one temperature observation at a time, generating a recurrent measurement of the temperature attributes as defined above. This way we changed from Log Enrichment to Log Extension. As apparent from Fig. 3 (second model, shaded area), the entailed activity loop has been process-mined correctly.

Running DPA this time, for each of the attributes, the very same classification is obtained, but with entirely different evaluation output. First of all, amazingly, the number of process instances increases erroneously to 130; this happens because, within a process instance, only the first of recurring events is used for subsequent decision analysis [4]: hence, in all of the 100 instances, the decision after the first temperature measurement (that is, the first occurrence within the loop) branches to "Continue Transport", and just 30 instances – later in the process – "Return to Parking Lot" at all. A closer look at the log data unveils that, in 10 of the instances, the first temperature observation, respectively, exceeds the threshold value – which explains the 10 instances classified wrong.

We conclude, modeling the process in either approach cannot resolve the shortcoming of representing recurrent measurements (generated through process *loops*; [4]) of attributes for DPA, as there is no way to preserve the temporal structure of these measurements properly.

We now develop a new process view for the next iteration of the $DPA^{TimeSeries}$ method, which concentrates on the process instances and their decisions whether to return or not. For this view we use the monitoring data now as main source. This leads to an analysis model for classification of time series data. Because of regular structure of monitoring time we stick to Separation of Data and keep all 12 measurements as vector of attributes, but understand it as regular time series. Accordingly to the time series understanding we start with an analysis of the trend behavior and use parallel coordinate plots in R for the visualization of the groups. The results are shown in Fig. 4. While the critical plot (left side) only shows sharp single tops we can see clearly that the return plot (right side) has high plateaus leading immediately to the conjecture that the decision about return to parking lot depends on the duration of temperature above a threshold. From a more detailed investigation with visual data mining tools we can determine the rule: the critical event is that the temperature remain above threshold for two consecutive events.

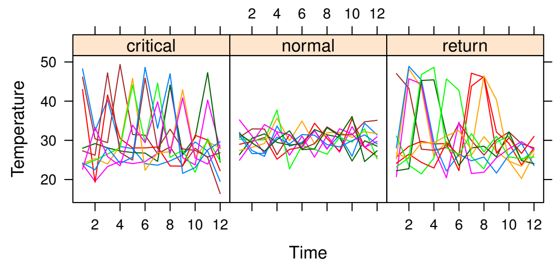


Fig. 4. Parallel Coordinate Plot of the all cases

One alternative now would be – using Log Extension again – to define a new event for the process which is defined as: "First occurrence of two consecutive measurements above the threshold". DPA would – that way – be able to identify this attribute as decisive.

We also applied different other classification methods for the data. It turned out that with Boosting and Support Vector Machines we obtained better results for the error rates in case of cross validation than with decision trees. But the results are not easy to interpret in application.

With that data understanding we now produce two aggregated data attributes: (i) a boolean attribute *temperatureThresholdViolation* indicating that the threshold we found using data mining was violated in two consecutive measurements; (ii) a numeric attribute *temperatureThresholdViolationCount* counting the number of these violations, bypassing the problem of loosing the temporal information. This way there is no need for the recurring events with single measurements and therefore we can again make use of the DPA by means of Log Enrichment using the two new aggregated attributes.

We start a new iteration of the $DPA^{TimeSeries}$ method with the augmented attribute set in the ProM environment and find with standard DPA 100% of cases are correctly classified.

4 Related Work

An integrated analysis of processes and data is provided by DPA [1], [4]. In [5], DPA was improved and generalized using algebraically-oriented procedures for finding complex decision rules with more than one variable. By contrast, the $DPA^{TimeSeries}$ method aims at finding new rules using statistically-oriented empirical methods, augmenting the space of possible decision functions with attributes through a data-driven search among empirical models. [6] overcomes other difficulties of DPA like invisible transitions and therefore certain kinds of loops within the process model or deviating behavior by control-flow alignment. Our approach differs from that in dealing with time series data and therefore recurring events that might not be found within existing log files. Our approach also resolves problems with loops through extending DPA with data mining techniques to identify aggregation value attributes and defining new events within the business processes these attributes can be attached to. Another interesting approach is [7] that addresses the clustering of health care processes. The $DPA^{TimeSeries}$, by contrast, focuses on the classification of temporal data occuring in connection with processes.

Log preparation tools cover the extraction and integration of data from different sources as well as data quality improvement, e.g., [8, 9]. Log enrichment is one possibility to deal with the latter, e.g. in [10] it is proposed to make more complex time data usable.

5 Conclusions

In this paper, we proposed the $DPA^{TimeSeries}$ method for analyzing time series data and process logs by a combined and iterative application of process and data mining techniques. For equipping and analyzing the logs with time series data, we discussed the possibilities of log enrichment and extension as well as of keeping log and time series data in a separated way. The $DPA^{TimeSeries}$ method is implemented and evaluated based on use case from the logistics domain.

References

1. Rozinat, A., van der Aalst, W.: Decision mining in ProM. In: Business Process Management. (2006) 420–425
2. Rinderle, S., Bassil, S., Reichert, M.: A framework for semantic recovery strategies in case of process activity failures. In: ICEIS (1). (2006) 136–143
3. Jensen, K., Kristensen, L.M., Wells, L.: Coloured petri nets and CPN tools for modelling and validation of concurrent systems. Int. J. Softw. Tools Technol. Transf. **9**(3) (2007) 213–254
4. Rozinat, A., van der Aalst, W.: Decision mining in business processes. Technical report (2006)
5. de Leoni, M., Dumas, M., Garcia-Banuelos, L.: Discovering branching conditions from business process execution logs. In: Fundamental Approaches to Software Engineering. (2013) 114–129
6. de Leoni, M., van der Aalst, W.M.: Data-aware process mining: discovering decisions in processes using alignments. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, ACM (2013) 1454–1461
7. Rebuge, Á., Ferreira, D.R.: Business process analysis in healthcare environments: A methodology based on process mining. Information Systems **37**(2) (2012) 99–116
8. Rodriguez, C., Engel, R., Kostoska, G., Daniel, F., Casati, F., Aimar, M.: Eventifier: Extracting process execution logs from operational databases. In: BPM 2012 Demo Track. (2012)
9. Nooijen, E.H.J., Dongen, B.F.v., Fahland, D.: Automatic discovery of data-centric and artifact-centric processes. In: Business Process Management Workshops. (2013) 316–327
10. Dunkl, R.: Data improvement to enable process mining on integrated non-log data sources. In Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A., eds.: Computer Aided Systems Theory - EUROCAST 2013. Volume 8111 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 491–498

Mining SQL Execution Traces for Data Manipulation Behavior Recovery

Marco Mori*, Nesrine Noughi, and Anthony Cleve

PReCISE Research Center, University of Namur
{marco.mori,anthony.cleve,nesrine.noughi}@unamur.be

Abstract. Modern data-intensive software systems manipulate an increasing amount of heterogeneous data in order to support users in various execution contexts. Maintaining and evolving activities of such systems rely on an accurate documentation of their behavior which is often missing or outdated. Unfortunately, standard program analysis techniques are not always suitable for extracting the behavior of data-intensive systems which rely on more and more dynamic data access mechanisms which mainly consist in run-time interactions with a database. This paper proposes a framework to extract behavioral models from data-intensive program executions. The framework makes use of dynamic analysis techniques to capture and analyze SQL execution traces. It applies clustering techniques to identify data manipulation functions from such traces. Process mining techniques are then used to synthesize behavioral models.

Keywords: data-manipulation behavior recovery, data-oriented process mining, data-manipulation functions

1 Introduction

Data-intensive systems typically consists of a set of applications performing frequent and continuous interactions with a database. Maintaining and evolving data-intensive systems can be performed only after the system has been sufficiently understood, in terms of structure and behavior. In particular, it is necessary to recover missing documentation (models) about the data manipulation behavior of the applications, by analyzing their interactions with the database. In modern systems, such interactions usually rely on dynamic SQL, where automatically generated SQL queries are sent to the database server.

The literature includes various static and dynamic program analysis techniques to extract behavioral models from traditional software systems. Existing *static* analysis techniques [19,18,22,7,20], analyzing program *source code*, typically fail in producing complete behavioral models in presence of dynamic SQL. They cannot capture the dynamic aspects of the program-database interactions, influenced by context-dependent factors, user inputs and results of

* beneficiary of an FSR Incoming Post-doctoral Fellowship of the *Académie universitaire 'Louvain'*, co-funded by the Marie Curie Actions of the European Commission

preceding data accesses. Existing *dynamic* analysis techniques [10], analyzing program *executions*, have been designed for other purposes than data manipulation behavior extraction. Several authors have considered the analysis of SQL execution traces in support to data reverse engineering, service identification or performance monitoring [8,9,12,11,23]. Such techniques look very promising for recovering an approximation of data-intensive application behavior.

In this paper, we propose a framework to recover the data manipulation behavior of programs, starting from SQL execution traces. Our approach uses clustering to group the SQL queries that implement the same high-level data manipulation function, i.e., that are syntactically equal but with different input or output values. We then adopt classical process mining techniques to recover data manipulation processes. Our approach operates at the level of a *feature*, i.e., a software functionality as it can be perceived by the user. A feature corresponds to a *process* enabling different instances, i.e., *traces*, each performing possibly different interactions with a database.

The reminder of this paper presents in Section 2 our approach along with a tool-supported validation. Finally, Section 3 discusses related work and Section 4 ends the paper showing possible future directions.

Motivating Example. We consider an e-commerce web store for selling products in a world-wide area. The system provides a set of features requiring frequent and continuous interactions with the database by means of executing SQL statements. For instance, the feature for retrieving products (*view_products*) accesses information about categories, manufacturers and detailed product information. Which data are accessed at runtime depends on dynamic aspects of the system. For example, given that a certain feature instance retrieves the categories of products before accessing product information we can derive that it corresponds to a category-driven search. If a certain instance accesses manufacturer information before product information we analogously derive that it corresponds to a manufacturer-driven search. By capturing and mining the database interactions of multiple feature instances, it is possible to recover the actual data manipulation behavior of the feature, e.g., a process model with a variability point among two search criteria.

2 Data Manipulation Behavior Recovery

Our framework supports the extraction of the data manipulation behavior of programs by exploiting several artifacts (see Fig. 1). We assume the existence of a *logical* and possibly of a *conceptual schema* with a mapping between them. The *conceptual schema* is a platform-independent specification of the application domain concepts, their attributes and relationships. The *logical schema* contains objects (tables, columns and foreign keys) implementing abstract concepts over which queries are defined. The conceptual schema and the mapping to the logical schema can be either available, or they can be obtained via database reverse engineering techniques [13]. Queries defined over the logical schema materialize the interactions occurring between multiple executions (traces) of a feature and the underlying database. Once the source code related to a feature has been identified [14], different techniques can capture SQL execution traces. Those techniques, compared in [9], range from using the DBMS log to sophisticated source code transformation. Among others, the approaches presented in [1,17] recover the link between SQL executions and source code locations through automated program instrumentation, while [6] makes use of tracing aspects to capture SQL execution traces without source code alteration. Once a sequence of queries is captured, it is necessary to identify the different traces, each corresponding to a feature instance. This problem has been tackled in the literature of specification mining by analyzing value-based dependencies of methods calls [3].

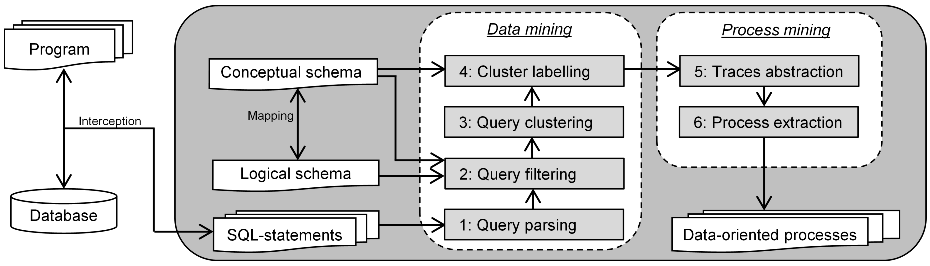


Fig. 1. Basics models: artifacts and components

Our approach is independent from the adopted trace capturing techniques. For each feature, it requires as minimal input a set of execution traces, each trace consisting of a sequence of SQL queries.

Query parsing (1). We characterize SQL queries according to (1) the information they recover or modify and (2) the related selection criteria. To this end, for each query we record a set of data-oriented properties according to the query type. For a *select* query we record a property with the *select* clause while for *delete*, *update*, *replace* or *insert* queries we record a property with the name of the table. If the query is either *update*, *replace* or *insert* we also record a property with the *set* clause and all its attributes. Finally for all query types but the *insert* we add a property for the *where* clauses along with their attributes. By means of these properties we ignore the actual values taken as input and produced as output by each query. Figure 2 shows three SQL traces along with their corresponding properties. For instance, query q_1 is a select query over attribute $Password$ of $Customer$ table (property p_1) and it contains a *where* clause with an equality condition over Id attribute (p_2); query q_3 is a *select* over attributes Id and $Price$ of $Product$ (property p_4), it contains two where clauses, i.e., a natural join between $Product.Id$ and $PCategory.Product\_Id$ (p_5) and an equality condition over $PCategory.Category\_Id$ attribute (p_6).

Query filtering (2). We remove from the input traces the queries that do not express end-user concepts, i.e., the ones referring to database system tables or log tables appearing only in the logical schema. In our example we remove q_{10}

Trace 1:

```
q1: SELECT Customer.Password FROM Customer WHERE Customer.Id = 'Mark27'; [p1,p2]
q2: SELECT Category.Id, Category.Image FROM Category; -> [p3]
q3: SELECT Product.Id, Product.Price FROM Product, PCategory WHERE Product.Id=PCategory.Product_Id AND
      PCategory.Category_Id='1'; -> [p4,p5,p6]
q4: SELECT PLang.Description FROM PLang, Language WHERE PLang.Language_Id=Language.Code AND PLang.Product_Id
      ='1A23' AND Language.Name='Italian'; -> [p7,p8,p9,p10]
q5: SELECT SpecialProduct.NewPrice FROM SpecialProduct,Product WHERE SpecialProduct.Product_Id=Product.Id
      AND Product.Id='1A23'; -> [p11,p12,p13]
q6: SELECT Manufacturer.Name FROM Manufacturer,Product WHERE Manufacturer.Id=Product.Manufacturer_Id AND
      Product.Id='1A23'; -> [p14,p15,p13]
q7: SELECT PLang.Description FROM PLang, Language WHERE PLang.Language_Id=Language.Code AND PLang.Product_Id
      ='1F32' AND Language.Name='Italian'; -> [p7,p8,p9,p10]
q8: SELECT SpecialProduct.NewPrice FROM SpecialProduct,Product WHERE SpecialProduct.Product_Id=Product.Id
      AND Product.Id='1F32'; -> [p11,p12,p13]
q9: SELECT Manufacturer.Name FROM Manufacturer,Product WHERE Manufacturer.Id=Product.Manufacturer_Id AND
      Product.Id='1F32'; -> [p14,p15,p13]
q10: INSERT INTO Log(IdEvent,Event,Date,Time) VALUES ('021','PrAcc1A23-1F32','2013-02-22','12:21:00'); -> [
      p16]
```

Trace 2:

```
q11: SELECT Customer.Password FROM Customer WHERE Customer.Id = 'JennyMa'; [p1,p2]
q12: SELECT Category.Id, Category.Image FROM Category; -> [p3]
q13: SELECT Product.Id, Product.Price FROM Product, PCategory WHERE Product.Id=PCategory.Product_Id AND
      PCategory.Category_Id='2'; -> [p4,p5,p6]
```

Trace 3:

```
q14: SELECT Customer.Password FROM Customer WHERE Customer.Id = 'DanWer'; [p1,p2]
q15: SELECT Manufacturer.Id, Manufacturer.Name FROM Manufacturer -> [p17]
q16: SELECT Product.Id, Product.Price FROM Product WHERE Product.Manufacturer_Id='AppleNamur01' -> [p4,p18]
q17: SELECT PLang.Description FROM PLang, Language WHERE PLang.Language_Id=Language.Code AND PLang.
      Product_Id='2D11' AND Language.Name='Italian'; -> [p7,p8,p9,p10]
q18: SELECT SpecialProduct.NewPrice FROM SpecialProduct,Product WHERE SpecialProduct.Product_Id=Product.Id
      AND Product.Id='2D11'; -> [p11,p12,p13]
q19: SELECT Manufacturer.Name FROM Manufacturer,Product WHERE Manufacturer.Id=Product.Manufacturer_Id AND
      Product.Id='2D11'; -> [p14,p15,p13]
q20: INSERT INTO Log(IdEvent,Event,Date,Time) VALUES ('022','PrAcc2D11','2013-02-28','14:00:03'); -> [p16]
```

SQL-statements properties:

```
p1="SELECT Customer.Password", p2="Customer.Id.EQ_VALUE", p3="SELECT Category.Id Category.Image",
p4="SELECT Product.Id Product.Price", p5="Product.Id=PCategory.Product_Id",
p6="PCategory.Category_Id.EQ_VALUE", p7="SELECT PLang.Description", p8="PLang.Language_Id=Language.Code",
p9="PLang.Product_Id.EQ_VALUE", p10="Language.Name.EQ_VALUE", p11="SELECT SpecialProduct.NewPrice",
p12="SpecialProduct.Product_Id=Product.Id", p13="Product.Id.EQ_VALUE", p14="SELECT Manufacturer.Name",
p15="Product.Manufacturer_Id=Manufacturer.Id",  p16="INSERT INTO Log",
p17=" SELECT Manufacturer.Id Manufacturer.Name", p18="Product.Manufacturer_Id.EQ_VALUE"
```

Fig. 2. Web Store: Traces of SQL statements with data-oriented properties

and q_{20} accessing table *Log* without a counterpart in the conceptual schema.
Query clustering (3). We cluster queries having the same data-oriented properties thus producing disjoint partitions, related to different database accesses. We report in Table 1 the clusters obtained from queries in Fig.2.

Table 1. Web Store: Clusters of SQL queries

C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**
$\{q_1, q_{11}, q_{14}\}$	$\{q_2, q_{12}\}$	$\{q_3, q_{13}\}$	$\{q_4, q_7, q_{17}\}$	$\{q_5, q_8, q_{18}\}$	$\{q_6, q_9, q_{19}\}$	$\{q_{15}\}$	$\{q_{16}\}$
$\{p_1, p_2\}$	$\{p_3\}$	$\{p_4, p_5, p_6\}$	$\{p_7, p_8, p_9, p_{10}\}$	$\{p_{11}, p_{12}, p_{13}\}$	$\{p_{13}, p_{14}, p_{15}\}$	$\{p_{17}\}$	$\{p_4, p_{18}\}$

Cluster labeling (4). We identify the data manipulation function implemented by each cluster by analyzing the conceptual schema fragment corresponding to the logical subschema accessed by the cluster queries. For determining the labels we adopt the same naming convection proposed in [5] to associate conceptual level operations to SQL query code. In addition, we associate the label with a set of input/output (I/O) parameters (see Table 2). Input parameters are the attributes involved in equality or inequality conditions that appear in the data-oriented properties of the queries, while output parameters are the set of attributes appearing within the *select* query property.

Table 2. Web Store: Clusters with data manipulation functions and I/O parameters

Cluster	**Input**	**Output**
C1:*getCustomerById*	$\{Id\}$	$\{Password\}$
C2*getAllCategory*	–	$\{Id, Image\}$
C3:*getAllProductOfCategoryViaPCategory*	$\{Category\_Id\}$	$\{Id, Price\}$
C4:*getAllLanguageOfProductViaPLang*	$\{Product\_Id, Name\}$	$\{Description\}$
C5:*getSpecialProductOfProductViaSProd*	$\{Product.Id\}$	$\{NewPrice\}$
C6:*getManufacturerOfProductViaPManufact*	$\{Product.Id\}$	$\{Name\}$
C7:*getAllManufacturer*	–	$\{Id, Name\}$
C8:*getAllProductOfManufacturerViaPManufact*	$\{Manufacturer\_Id\}$	$\{Id, Price\}$

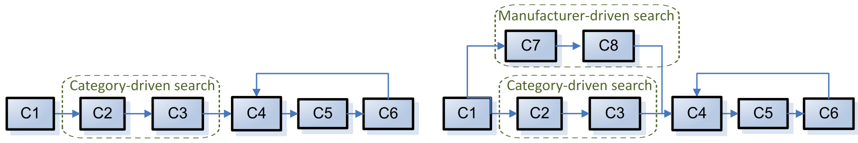
Process mining (5-6). We generate a process starting from a set of SQL traces of a single feature. The *traces abstraction* phase replaces SQL traces with the corresponding traces of data manipulation functions. The *process extraction* phase exploits a process mining algorithm to extract the feature behavior as a sequence of function executions with sequential, parallel and choice operators. In the following we show how to recover the data manipulation behavior of the *view_products* web-store feature starting from the traces of data manipulation functions in Table 3 (corresponding to queries in Fig.2).

Table 3. Web Store: Traces of data manipulation functions

Trace 1	*getCustomerById*(**C1**) - *getAllCategory*(**C2**) - *getAllProductOfCategoryViaPCategory*(**C3**) - *getAllLanguageOfProductViaPLang*(**C4**) - *getSpecialProductOfProductViaSProd*(**C5**) - *getManufacturerOfProductViaPManufact*(**C6**) - *getAllLanguageOfProductViaPLang*(**C4**) - *getSpecialProductOfProductViaSProd*(**C5**) - *getManufacturerOfProductViaPManufact*(**C6**)
Trace 2	*getCustomerById*(**C1**) - *getAllCategory*(**C2**) - *getAllProductOfCategoryViaPCategory*(**C3**)
Trace 3	*getCustomerById*(**C1**) - *getAllManufacturer*(**C7**) - *getAllProductOfManufacturerViaPManufact*(**C8**) - *getAllLanguageOfProductViaPLang*(**C4**) - *getSpecialProductOfProductViaSProd*(**C5**) - *getManufacturerOfProductViaPManufact*(**C6**)

Trace 1 gets customer information (C1), it performs a category-driven search of products by means of getting all the product categories (C2) and all the products of a certain selected category (C3). For each retrieved product, three functions are iterated: C4 retrieves product description, C5 extracts special product information and C6 extracts related manufacturer information. *Trace 2* is different from *Trace 1* because after function C3 no products are retrieved and the process ends. If we apply a mining algorithm to *Trace 1* and *2* we obtain a process (Fig. 3(a)) which performs consecutively functions C1, C2 and C3 before entering in the loop iterating C4, C5, and C6. The process ends after zero, one or more iterations of the loop. Let us now assume to include into the process *Trace 3* which is equal to *Trace 1* except that it searches products based on their manufacturer (functions C7 and C8) instead of searching by category (C2 and C3).

If we mine the process model by considering as input all the traces (Fig. 3(b)), we end up with a new alternative branch: the customer can now perform either a manufacturer-driven search or a category-driven search.



(a) (b)

Fig. 3. Web Store: process mined with (a) *Trace 1* and *2* and (b) *Trace 1*, *2* and *3*.

Tool support. The presented approach is implemented into an integrated tool which takes as input a set of SQL traces (each representing an instance of the same feature), the logical schema and optionally the conceptual schema and the conceptual-to-logical schema mapping. A SQL *parser* extracts the data-oriented properties while a *clustering* component exploits the *colibri-Java* Formal Concept Analysis tool[1] to cluster queries according to those properties. A *labeling* component generates data manipulation functions (i.e., cluster signatures) while a *trace abstraction* component uses a Java library[2] to create standardized event logs. Finally we rely on the de-facto standard process mining tool (*ProM* tool[3]) to create a Petri net from standardized event logs. *ProM* supports different process mining algorithms providing different trade-offs between completeness and noise [4] to be chosen according to specific application needs.

We applied our tool together with *ProM* and the ILP miner algorithm [21] (complete models with low noise) to extract data-oriented processes of a e-restaurant web application and we conducted a set of preliminary experiments to assess the sensitivity of our technique in producing correct processes depending on the traces log coverage. The tool supported the identification of correct features processes in a semi-automatic manner along with the help of the designer. A complete list of SQL statements grouped by feature with different traces, extracted data manipulation functions and corresponding processes are publicly available at the companion website[4]. The conceptual and logical schemas, accessible through the DB-MAIN[5] CASE tool, are also provided.

3 Related Work

In the literature different approaches use dynamic analysis of SQL queries with a different goal than data manipulation behavior understanding. The approaches

[1] `http://code.google.com/p/colibri-java/`
[2] `http://www.xes-standard.org/openxes/start`
[3] `http://www.promtools.org/`
[4] `http://info.fundp.ac.be/~mmo/MiningSQLTraces`
[5] DB-MAIN official website, `http://www.db-main.be`

presented in [8,9] analyze SQL statements in support to database reverse engineerinf, e.g., detecting implicit schema constructs [9] and implicit foreign keys [8]. The approach presented by Di Penta et al. [12] identifies services from SQL traces. The authors apply FCA techniques to name services I/O parameters thus supporting the migration towards Service Oriented Architecture. Debusmann et al. [11] present a dynamic analysis method for system performance monitoring, i.e., measuring the response time of queries sent to a remote database server. Yang et al. [23] support the recovery of a feature model by means of analyzing SQL traces. Although the former approaches analyze (particular aspects of) the data access behavior of running programs, none of the former approaches [8,9,12,11,23] is able to produce process models expressing such a behavior at a high abstraction level, as we do in this paper.

Other approaches (e.g., [16,15]) extract business processes by exploiting/combining static and dynamic analysis techniques, but they are not designed to deal with dynamically generated SQL queries. The most related approach, by Alalfi et al. [2], extracts scenario diagrams and UML security models by considering runtime database interactions and the state of the PHP program. These models are used for verifying security properties but they do not describe the generic data manipulation behavior of the program, they only analyze web-interface interactions. In addition they have not considered different possible instances of a given scenario as we claim it is necessary to extract a complete and meaningful model. Understanding processes starting from a set of execution traces is at the core of process mining. This paper does not make any additional contributions as far as process mining is concerned, but it is the first to apply such techniques to analyze program-database interactions.

4 Conclusion and future work

Our paper presented a tool-supported approach to recover the data manipulation behavior of data-intensive systems. The approach makes use of clustering, conceptualization and process mining techniques starting from SQL execution traces captured at runtime. The approach is independent from the type of systems considered, provided that a query interception phase is possible. It could, for instance, be applied to legacy cobol systems, Java systems with or without Object-Relational-Mapping technologies, or web applications written in PHP. As for future work we plan to enrich the input traces with multiple sources of information like user input, source code and queries results with the aim of identifying the conditions that characterize decision points within process models.

References

1. M. Alalfi, J. Cordy, , and T. Dean. WAFA: Fine-grained dynamic analysis of web applications. In *WSE 2009*, pages 41–50, 2009.
2. M. H. Alalfi, J. R. Cordy, and T. R. Dean. Recovering role-based access control security models from dynamic web applications. In *ICWE*, pages 121–136. 2012.

3. G. Ammons, R. Bodík, and J. R. Larus. Mining specifications. In *ACM Sigplan Notices*, volume 37, pages 4–16, 2002.
4. J. Buijs, B. Dongen, and W. Aalst. On the role of fitness, precision, generalization and simplicity in process discovery. In *OTM*, volume 7565 of *LNCS*, pages 305–322, 2012.
5. A. Cleve, A.-F. Brogneaux, and J.-L. Hainaut. A conceptual approach to database applications evolution. In *ER*, pages 132–145, 2010.
6. A. Cleve and J.-L. Hainaut. Dynamic analysis of SQL statements for data-intensive applications reverse engineering. In *WCRE 2008*, pages 192–196, 2008.
7. A. Cleve, J. Henrard, and J.-L. Hainaut. Data reverse engineering using system dependency graphs. In *WCRE 2006*, pages 157–166, 2006.
8. A. Cleve, J.-R. Meurisse, and J.-L. Hainaut. Database semantics recovery through analysis of dynamic sql statements. *J. Data Semantics*, 15:130–157, 2011.
9. A. Cleve, N. Noughi, and J.-L. Hainaut. Dynamic program analysis for database reverse engineering. In *GTTSE*, pages 297–321, 2011.
10. B. Cornelissen, A. Zaidman, A. van Deursen, L. Moonen, and R. Koschke. A systematic survey of program comprehension through dynamic analysis. *IEEE Trans. Software Eng.*, 35(5):684–702, 2009.
11. M. Debusmann and K. Geihs. Efficient and transparent instrumentation of application components using an aspect-oriented approach. In *DSOM 2003*, volume 2867 of *LNCS*, pages 227–240, 2003.
12. C. D. Grosso, M. D. Penta, and I. G. R. de Guzmán. An approach for mining services in database oriented applications. In *CSMR*, pages 287–296, 2007.
13. J.-L. Hainaut, J. Henrard, V. Englebert, D. Roland, and J.-M. Hick. Database reverse engineering. In *Encyclopedia of Database Systems*, pages 723–728. Springer US, 2009.
14. H. Kazato, S. Hayashi, T. Kobayashi, T. Oshima, S. Okada, S. Miyata, T. Hoshino, and M. Saeki. Incremental feature location and identification in source code. In *CSMR*, pages 371–374, 2013.
15. Y. Labiche, B. Kolbah, and H. Mehrfard. Combining static and dynamic analyses to reverse-engineer scenario diagrams. In *ICSM*, pages 130–139, 2013.
16. H. R. M. Nezhad, R. Saint-Paul, F. Casati, and B. Benatallah. Event correlation for process discovery from web service interaction logs. *VLDB*, 20(3):417–444, 2011.
17. M. N. Ngo and H. B. K. Tan. Applying static analysis for automated extraction of database interactions in web applications. *Information and software technology*, 50(3):160–175, 2008.
18. J.-M. Petit, J. Kouloumdjian, J.-F. Boulicaut, and F. Toumani. Using queries to improve database reverse engineering. In *ER*, pages 369–386, 1994.
19. J. C. Silva, J. C. Campos, and J. Saraiva. Gui inspection from source code analysis. *ECEASST*, 33, 2010.
20. H. van den Brink, R. van der Leek, and J. Visser. Quality assessment for embedded sql. In *SCAM*, pages 163–170, 2007.
21. J. M. E. van derWerf, B. F. van Dongen, C. A. Hurkens, and A. Serebrenik. Process discovery using integer linear programming. *Fundamenta Informaticae*, 94(3):387–412, 2009.
22. D. Willmor, S. M. Embury, and J. Shao. Program slicing in the presence of a database state. In *ICSM 2004*, pages 448–452, 2004.
23. Y. Yang, X. Peng, and W. Zhao. Domain feature model recovery from multiple applications using data access semantics and formal concept analysis. In *WCRE*, pages 215–224, 2009.

A Broker Framework for Secure and Cost-Effective Business Process Deployment on Multiple Clouds

Elio Goettelmann[1,2], Karim Dahman[3], Benjamin Gateau[2] and Claude Godart[1]

[1] LORIA - INRIA Grand Est, Université de Lorraine, Nancy, France.
[2] CRP Henri Tudor, Kirchberg, Luxembourg.
[3] Blu Age - Netfective Technology, Pessac, France.
elio.goettelmann@tudor.lu, k.dahman@bluage.com, benjamin.gateau@tudor.lu, claude.godart@loria.fr

Abstract. Security risk management on information systems provides security guarantees while controlling costs. But security risk assessments can be very complex, especially in a cloud context where data is distributed over multiple environments. To prevent costs from becoming the only cloud selection factor, while disregarding security, we propose a method for performing multiple cloud security risk assessments. In this paper we present a broker framework for balancing costs against security risks. Our framework selects cloud offers and generates deployment-ready business processes in a multi-cloud environment.

Keywords: Business Process, Security Risk Management, Cloud

1 Introduction

The Cloud business model proposes a multitude of different services, at different prices, and with various quality levels. While the use of cloud computing can reduce costs, the selection of a solution is time consuming. For this purpose, cloud brokers have emerged; they can help cloud consumers to select adequate solutions, by comparing existing offers, essentially against their prices.

But security is still an important factor for a cloud selection process. As cloud computing presents new kinds of security risks ([3], [5]), they need to be treated before wider adoption. Novel methods have to be defined in order to prevent these potential losses on companies.

In turn, distributing software over multiple locations increases the complexity of gathering sensitive business information. In this paper, we propose a framework for cloud brokers which helps them analyze the security levels of different cloud offers, following standard risk assessment methodologies, with respect to cloud offers.

The paper is organized as follows. Section 2 presents a motivating example used to demonstrate the purpose and the scope of our framework. Section 3 describes our tool, its implementation on the motivating example and experimental results. Section 4 and Section 5 discuss respectively related and future work.

2 Motivating Example and Overview

In this section, we introduce a motivating example to illustrate our framework. Then we give an overview of our approach for selecting clouds when deploying business processes.

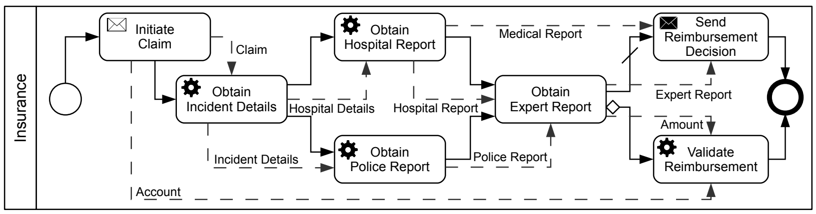


Fig. 1. Business Process Motivation Example: Insurance Claim Recovery Chain.

Consider an *insurance claim recovery chain* [7] as a BPMN business process model depicted in Fig. 1. This process is initiated when an *insurance* company receives a claim recovery declaration from a *beneficiary*. To obtain details about the incident, the *emergency* service is invoked. The *hospital* and *police* reports are required by the *expert* to decide if the *reimbursement* will be accepted or not. If so, the *bank* is requested and a notification is sent to the *beneficiary*.

Now suppose that the insurance company wants to outsource the software supporting this process to the cloud to reduce costs. It has not necessarily the knowledge of moving it effectively on its own. A cloud broker could support the company by **evaluating the security risks** of a cloud outsourcing, **selecting the adequate offers** and **decomposing the process** to deploy it on multiple clouds. These three tasks are detailed below.

Our tool consists in a design-time framework for producing secure and cost-effective business processes on multiple cloud. It is illustrated in Fig.2.

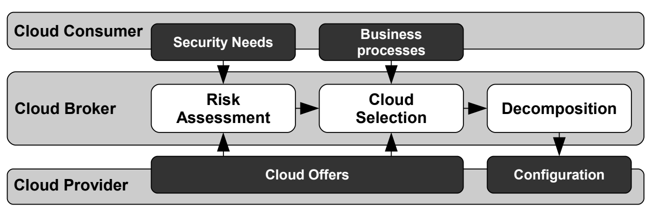


Fig. 2. Our design-time framework for multi-cloud business process deployment.

First, a cloud broker requests the *security needs* as non-functional requirements from the cloud consumer and analyzes the *cloud offers* from the cloud provider to realize a **risk assessment**.

Second, the broker uses the *business processes* from the cloud consumer to **select** the adapted *cloud offers* based on the functional requirements, the costs and the previously calculated risk.

Third, the broker **decomposes** the business process into smaller parts, as each task can be enacted on a different cloud offer. The generated *configuration* is the assignment of these process fragments to cloud offers. The decomposition itself has already been addressed in [8].

3 Implementation, experimentation and evaluation

To demonstrate the feasibility of the approach, we illustrate on our motivating example the three previously defined steps of the broker tool.

3.1 Risk Assessment

The risk assessment is threefold: security needs definition, risk evaluation and cloud provider exclusion.

Security needs definition Our framework uses the five CIANA objectives (*Confidentiality, Integrity, Availability, Non-Repudiation, Authenticity*) to define the *security need* of each data object of the process (Fig. 1). If the data object needs the objective, the value is equal to 1, otherwise to 0 (see Table 1). Typically, these security needs are negotiated by a risk manager with the cloud consumer.

Table 1. Annotations of the security objectives on the motivation example.

Data Associations	Conf.	Integ.	Avail.	N.-rep.	Auth.
Claim	0	0	1	1	0
Hospital details	0	1	1	0	1
Incident details	0	1	1	0	1
Hospital report	1	1	0	1	1
Police report	1	1	0	1	1
Medical report	1	1	0	1	1
Expert report	1	1	0	1	1
Account	1	1	0	0	1
Amount	1	1	0	1	1

As business process deployment is task-centric, these values need to be translated into security needs on tasks. We use the maximum values of the input and output objects of a task. For example, *Obtain Incident Details* is associated to the data objects *Claim*, *Incident Details* and *Hospital Details*. So, by taking the maximum of each data object's need we get the need of *Obtain Incident Details*: {0, 1, 1, 1, 1} for respectively $\{Confidentiality$, *Integrity*, *Availability*, *Non-Repudiation*, *Authenticity*}.

Risk evaluation for each provider In this paper we take into account five cloud security threats given by the CSA [3] to evaluate the risk. These threats are each related to the 5 CIANA objectives:

- *Data Breaches* = {*Confidentiality*}
- *Data Loss* = {*Availability, Non-Repudiation*}
- *Account Hijacking* = {*Confidentiality, Integrity, Availability, Non-Repudiation, Authenticity*}
- *Insecure Interfaces* = {*Confidentiality, Integrity, Authenticity*}
- *Denial of Service* = {*Availability*}

Harm - We combine these relations with the security needs of each data object to obtain a so-called **harm**. The **harm** is defined on each data object, for each threat through the sum of the affected security needs. For example, the *Insecure Interfaces* threat (t) has the following harm on *Obtain Incident Details* (ta):
$Harm(t, ta) = (1 \times 0) + (1 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 1) = 3$

The first value of each bracket is equal to 1 if the threat is related to the objective, 0 otherwise, and the second value is the need calculated previously. Notice that the harm value is in this case always between 0 and 5. The result for all tasks of the process can be seen in Table 2.

Coverage - Now that we have the harm of a threat on each task of the process, we need to determine the response to these threats for each cloud provider. For this information we use the STAR Registry and the matrix defined by the CSA [3].

The CSA matrix defines a list of security controls a cloud provider should implement to reduce security risks. Each of these controls can be related to one or multiple threats. For example, the control "*IS-19.4 - Do you maintain key management procedures ?*" mitigates the *Data Breaches* threat. This matrix defines a total of 197 controls to implement in order to respond to all threats in the best possible way.

The STAR Registry publishes the list of implemented controls for providers willing to follow these recommendations. These lists are freely accessible and can help to check if a control has been put in place by a specific provider or not.

In our case, we use these two information as binary values (a control mitigates a threat or not / a control is implemented by a provider or not) to calculate a so-called **coverage score**, which indicates the response of a provider to a given threat. This value is a percentage, if the provider implements all controls mitigating a threat, it gets a coverage for this threat of 100%. In our case, this percentage is brought to a score on a scale of 0 to 5 (with 5 equivalent to 100%). We took 5 providers from the registry and calculated their coverage score, it is available in Table 2.

Table 2. Harms on the process tasks and coverage of the providers for 5 cloud threats

	Initiate claim	Obt. incident det.	Obt hospital rep.	Obt. police rep.	Obt. expert rep.	Send reimb. dec.	Process reimb.	Softlayer	CloudSigma	FireHost	SHI Intern.	Terremark
	Harm							Coverage				
Data breaches	1	0	1	1	1	1	1	2	5	2	4	4
Data loss	2	2	2	2	1	1	1	3	4	2	4	5
Account Hijacking	5	4	5	5	4	4	4	4	5	3	5	3
Insecure interfaces	3	2	3	3	3	3	3	4	5	3	4	4
Denial of service	1	1	1	1	0	0	0	4	5	4	5	5

Too risky provider exclusion Usually, the *vulnerability* is assessed and used to calculate a risk value of an information system. But in a cloud context, providers may be tempted to conceal their vulnerabilities for security reasons. This is why we use the **coverage** based on the security controls. By using the maximum possible coverage value $Covg_{max}$ (in our case 5), it is possible to get an equivalent to the vulnerabilities. Therefore, by combining this value with the **harm** we can define the following risk formula for a threat t, a task ta and a provider p:
$Risk(t, ta, p) = Harm(t, ta) + (Covg_{max} - Covg(p, t))$

Table 3 shows the maximum risk value for the five CSA threats, for the tasks of our motivating example, on five different providers (*Softlayer*[4], *CloudSigma*[5], *FireHost*[6], *SHI Int.*[7] and *Terremark*[8]).

Table 3. Maximum risk value of the tasks for each provider

	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim	6	5	7	5	7
Obt. incident det.	5	4	6	4	6
Obt. hospital rep.	6	5	7	5	7
Obt. police rep.	6	5	7	5	7
Obt. expert rep.	5	4	6	4	6
Send reimb. dec.	5	4	6	4	6
Process reimb.	5	4	6	4	6

In accordance with the consumer, the broker defines the level of acceptable risk (referred to as **threshold**). For a given task, this threshold defines the providers with a too high risk value and excludes these deployment options. In our example, we set the threshold to 5, the cells of eliminated providers are grayed out in Table 3. Respectively a white cell means that the task can be deployed on the provider.

3.2 Cloud Selection

We select the target cloud environments in two stages: different configurations evaluation and final clouds selection.

Configurations evaluation We need to evaluate the different possible deployment configurations. To do this, we introduce a cost model which will allow us to balance the risks against the costs.

Cost model - Our cost model takes into account three types of costs:

- **Usage costs**, the CPU power needed to execute the process ($/GHz/month). The need is annotated on the tasks of the process.
- **Storage costs**, the space needed by the data of the process ($/GB/month). The size is annotated on the data objects of the process.
- **Transfer costs**, the amount of incoming/outgoing messages ($/GB). This size is calculated with the data exchanged between the process fragments.

Table 4 gives costs for the selected cloud offers of our motivating example.

[4] http://www.softlayer.com
[5] http://www.cloudsigma.com
[6] http://www.firehost.com
[7] http://www.shi.com
[8] http://www.terremark.com

Table 4. Costs of 5 Cloud offers

	Usage ($/GHz/mo)	Storage ($/GB/mo)	Transfer ($/GB)
Softlayer	20.00	0.10	0.10
CloudSigma AG	13.86	0.18	0.06
FireHost	25.70	2.78	0.50
SHI International, Corp.	11.56	0.29	0.01
Terremark	3.60	0.25	0.17

To find a deployment configuration (which task will be enacted on which cloud) we use an heuristic approach described in [6].

For our motivating example we tested our algorithm in three different ways. The results are shown in Table 5.

First run has no restrictions regarding the risk value, only the costs are taken into account. This gives us a "cheap" solution while leaving out security.

Second run includes a global risk threshold of 6. The majority of the tasks are now located on a more expensive but also less "risky" offer.

Third run has a global risk threshold set to 6. Once again, the solution becomes more expensive, but could be considered more "secure" than the second run.

We can notice that the **transfer costs** conduct to a regrouping of the tasks on one main offer to restrain the global costs.

Table 5. Output for different runs

	First run					Second run					Third run				
	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark	Softlayer	CloudSigma	FireHost	SHI Int.	Terremark
Initiate claim					x				x			x			
Obt. incident det.					x					x				x	
Obt. hospital rep.				x					x			x			
Obt. police rep.					x				x			x			
Obt. expert rep.					x				x			x			
Send reimb. dec.					x				x			x			
Process reimb.					x				x			x			
Risk	7					6					5				
Cost ($/mo)	84.58					206.68					259.24				

For those tests we optimized the costs while constraining the risk, but the algorithms can be extended to optimize the risk and constrain the cost, or even take into account other criteria.

Final configuration selection These resulting deployment configurations are analyzed by the cloud broker in conjunction with the cloud consumer to select the most adequate one. For our motivating example, we select the *Third run*.

3.3 Process deployment in the cloud

The process is deployed on the target environments in two steps: process decomposition and fragments deployment.

Process decomposition According to the selected configuration, the process is decomposed in multiple process fragments. A fragment is an autonomic business process enacted on one cloud and includes additional *synchronization tasks*. These additional tasks support the collaboration with the remote fragments to guarantee the control flow of the initial process. More details can be accessed in [7] and in [8].

The decomposed motivating example according to the selected configuration is depicted in Fig.3 (grey activities are *synchronization tasks*).

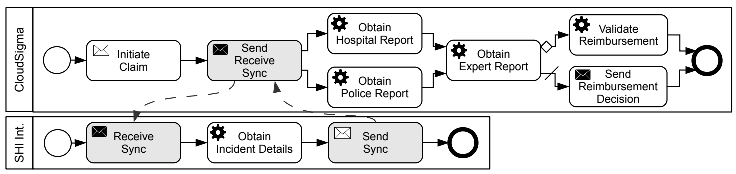


Fig. 3. A partitioned orchestration process.

Deployment in clouds The last step of our approach consists in deploying this configuration on the selected cloud offers. In [8] we presented how we deploy such service composition as BPEL programs on remote service orchestration engines (e.g., Apache ODE[9]).

4 Related Work

In [9] the authors present methods to distribute applications on different cloud environments. However security aspects are not considered. The authors of [2] are adapting processes through risk-reduction patterns, and in [13] processes are analyzed to decide if they are ready for a cloud deployment or not. But these two methods do not show the calculation of the risk value and do not consider process fragmentation.

In opposition to our proposal, [4] provides a risk-prediction algorithm to help users to take decisions during the execution of the process. We focus on design-time rather than on runtime, which changes slightly the kind of treated risks.

The model presented in [11] allows to evaluate security vulnerabilities in a Service Oriented Architecture, but does not take into account the cloud context. Our coverage approach based on security controls given in different standards ([1], [3]) seems to be more adapted to such a context.

Watson [12] decomposes workflows and deploys them on multiple clouds according to a cost model, but he defines arbitrary security levels for each provider. A more complete cost model is presented in [10], but it is not easily adaptable in the business process context for an automated treatment as it is done in our approach.

[9] `http://ode.apache.org/`

5 Conclusion and Future Work

In this paper we have presented a cloud broker framework for assessing security risks in a multiple-cloud context. We assess security risks of cloud providers using standard-based and industry accepted security controls and risk listings. We focus on one business process to illustrate how these risk values, in combination with costs, can help a cloud broker to take decisions for the cloud provider selection. The paper demonstrates the feasibility of our approach with a motivating example and real cloud providers.

Some limitations are not addressed in this paper. First, the shortage of empirical evaluation on real use cases, which will be realized in future works with domain experts and industrial partners. Another point is that our approach takes place at design-time, but as the Cloud is a very dynamic context, extending our framework to configuration at run-time would be an interesting improvement. Finally, the binary values for the *security needs*, *mitigations* and *control implementation* could be replaced with more complete scales, as some security controls "better" mitigate threats than others.

References

1. ISO/IEC 27017, Information tech., Security techniques, Code of practice for information security controls for cloud computing services based on ISO/IEC 27002.
2. O. Altuhhova, R. Matulevicius, and N. Ahmed. Towards definition of secure business processes. In *CAiSE Workshops*, pages 1–15, 2012.
3. Cloud Security Alliance. Cloud Control Matrix / Security, Trust & Assurance Registry / Consensus Assessments Initiative Questionnaire. Technical report.
4. R. Conforti, M. de Leoni, M. L. Rosa, and W. M. van der Aalst. Supporting risk-informed decisions during business process execution. In *CAiSE'13*, pages 116–132, Valencia, Spain, 2013.
5. European Network and Information Security Agency. Benefits, risks and recommendations for information security. Technical report, 2009.
6. W. Fdhila, M. Dumas, and C. Godart. Optimized decentralization of composite web services. In *CollaborateCom'10*, pages 1–10, 2010.
7. W. Fdhila, U. Yildiz, and C. Godart. A flexible approach for automatic process decentralization using dependency tables. In *ICWS '09*, pages 847–855, Washington, DC, USA, 09. IEEE Computer Society.
8. E. Goettelmann, W. Fdhila, and C. Godart. Partitioning and cloud deployment of composite web services under security constraints. In *IC2E'13*, 2013.
9. F. Leymann, C. Fehling, R. Mietzner, A. Nowak, and S. Dustdar. Moving applications to the cloud: an approach based on application model enrichment. *IJCIS*, 20(3):307–356, 2011.
10. B. Martens, M. Walterbusch, and F. Teuteberg. Costing of cloud computing services: A total cost of ownership approach. In *ICSS'12*, pages 1563–1572, 2012.
11. S. Sackmann, L. Lowis, and K. Kittel. A risk based approach for selecting services in business process execution. In *Wirtschaftsinformatik (1)*, pages 357–366, 2009.
12. P. Watson. A multi-level security model for partitioning workflows over federated clouds. In *CloudCom*, pages 180–188, 2011.
13. S. Wenzel, C. Wessel, T. Humberg, and J. Jürjens. Securing processes for outsourcing into the cloud. In *CLOSER*, pages 675–680, 2012.

A Method for Eliciting Security Requirements from the Business Process Models

Naved Ahmed and Raimundas Matulevičius

Institute of Computer Science, University of Tartu
J. Liivi 2, 50409 Tartu, Estonia
{naved, rma}@ut.ee

Abstract. In recent years, the business process modelling is matured towards expressing enterprise's organisational behaviour (i.e., business values and stakeholder interests). This shows potential to perform early security analysis to capture enterprise security needs. Traditionally, security in business processes is addressed either by representing security concepts graphically or by enforcing these security constraints. However, these approaches miss the elicitation of security needs and their translation to security requirements for system-to-be. This paper proposes a method to elicit security objectives from business process models and translate them to security requirements. The method enables early security analysis and allows developers not only to understand how to protect secure business assets, but it also contributes to alignment of the business processes with the technology that supports the execution of the business processes.

Keywords: Security in Business Processes, Business Process Modelling, Requirements Engineering

1 Introduction

Although the importance of introducing security engineering practices early in the development cycle has been acknowledged, it has been overssighted in business processes and targets the improvement of business function. The reason behind is that the business analysts are expert in their domain but having no clue about the security domain [10]. There has been several attempts to engage the relatively matured security requirements engineering in business processes. The majority of studies either focusses on the graphical representation of security aspects in business process models [8,10] or enforces the security mechanisms [7] or both [11]. These studies have neglected the security requirements elicitation. They analyse major problems when addressing security engineering in business process modelling. Firstly, security requirements are specified in terms of security architectural design (i.e., security control) and missing the rationale about the trade-offs of the security decision. Secondly, the requirement elicitation is either missing or haphazard: this leads to miss some critical security requirements. And finally, due to the dynamic and complicated nature of business processes the studies only addresses varying aspects (i.e., authorization, access control, separation of duty or binding of duty) but not the overall security of business processes. These problems can be overcome by eliciting security objectives from the organizational business processes and by transforming them to the security requirements of the operational business processes where the technology supports the business processes execution.

Here we analyse the research question, *how to elicit security objectives from the business processes and to translate them to security requirements*? We propose a method consisting of two major stages. Firstly, it describes how to identify business assets and to determine their security objectives. Secondly, it supports eliciting security requirements from the operational business processes.

The rest of the paper is structured as follows. In Section 2 we introduce the illustrative example. Section 3 presents our method for eliciting security requirements from the business processes. Section 4 concludes the paper and presents some future work.

2 Land Management System

To perform the security requirements elicitation one needs to collect the knowledge of enterprise *value system*, including the *value chain* and the *business functions*. Fig. 1 illustrates a value chain for the LMS example. It organises the enterprise business functions and relates them to each other (as enterprise cooperates to achieve the business goals). In Fig. 2 we present a detailed workflow of Prepare Plan *p*rocess. The *p*rocess has two business partners (Lodging Party and Planning Portal) expressed as *s*wimlanes, while Registry is identified as an information system.

Fig. 1: Land Management Systems - Value Chain

Similarly to Prepare Plan, other sub-processes (e.g., Lodge Plan, Examine Plan, Approve Plan and Update Plan) are also expanded to the operational and conversation models. But in Section 3, we will present our proposal using the Prepare Plan process (as illustrated in Fig. 1 and 2).

3 Security Requirement Elicitation Method

In [2], we have presented a set of security risk-oriented patterns for securing business processes. Based on these patterns, in this section, we introduce a method (Fig. 3) to elicit security requirements as constraints that have to be respected when executing a business process. The first stage is dedicated to *business asset identification* and *security objective determination*. In the second stage, the *elicitation of security requirements* is done from the system's contextual areas.

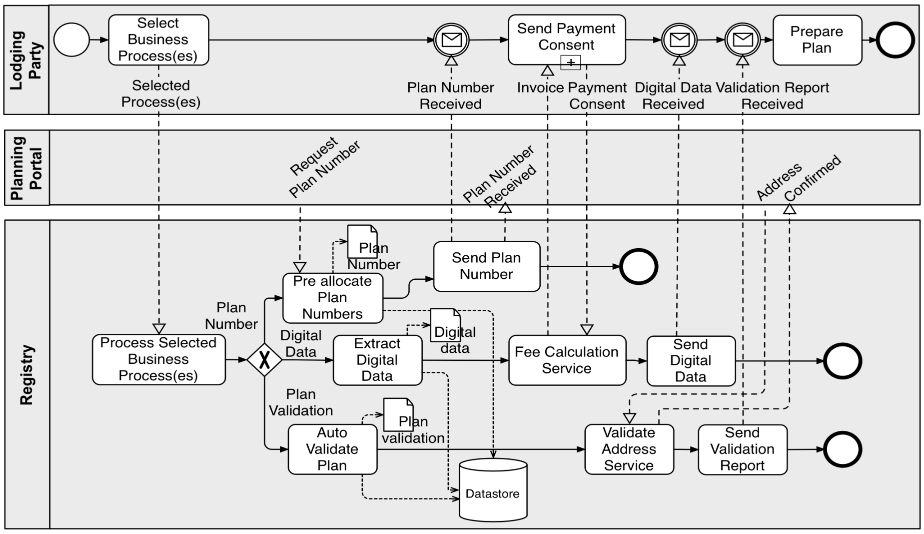


Fig. 2: Operational Business Process - Prepare Plan

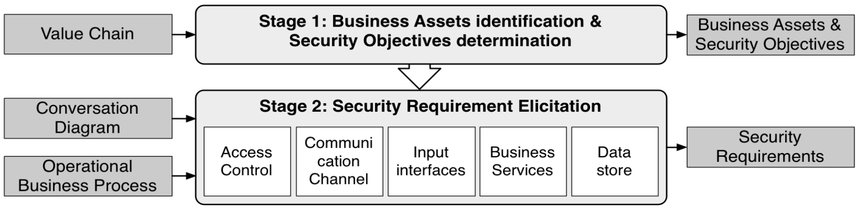


Fig. 3: Security Requirements Elicitation Method

3.1 Stage 1: Business Assets identification & Security Objectives determination

The first stage starts with the analysis of the *value chain*, which (i) gives an understanding of organisational processes and, thus, (ii) helps determine the assets that must be protected against security risks. In the LMS case the protected asset is Plan since it is the central artefact used in all the business activities (see Fig. 1). In terms of the security objective: i) Plan should be confidential, i.e., no unauthorised individual should read it and its relevant data; ii) Plan should be integral, i.e., the Plan and its relevant data should not be tempered; and iii) Plan and its relevant data should be available to the business partners at anytime.

3.2 Stage 2: Security Requirement Elicitation

At the second stage, the security requirements elicitation is performed at five contextual areas: *access control*, *communication channel*, *input interfaces*, *business services*, and *data store*. It is important to note that each artefact –*data* or *process*– separately considered and protected at each contextual area, contributes to the security of business asset (i.e., Plan) identified at the first stage.

Access Control specifies how the business assets could be manipulated by individuals, applications or their groups. The major concern is to protect the confidentiality of identified business asset, in our example the Plan, when it is being manipulated by the IS asset, (i.e., the Registry). The security threat arises if the access to the Plan and its properties, like (Plan Number, Digital Data, and Plan Validation) is allowed to users without checking their access permissions. The risk event would: *i*) negate confidentiality of Plan, *ii*) lead to the Plan unintended use, and *iii*) harm the Registry's reliability.

A way to mitigate the security risk is the introduction of access control mechanism, for example the Role-Based Access Control (RBAC) model. A *r*ole (e.g., Lodging Party and Planning Portal modelled using ≪role≫ stereotype) is a job function within the context of organisation. *P*ermissions characterise role privileges to perform operations on the protected object. An *o*bject is a protected resource (i.e., Plan). An *o*peration is an executable set of actions that can change the state of the protected resource. For instance, Pre allocate Plan Numbers, Send Digital Data, etc are operations which manipulate properties Plan Number, Digital Data and Plan Validation (Fig. 4). Permissions specify the security actions –namely, *Create*, *Read* and *Update*– that the role can perform over the state of the protected resource. The following security requirements are defined:

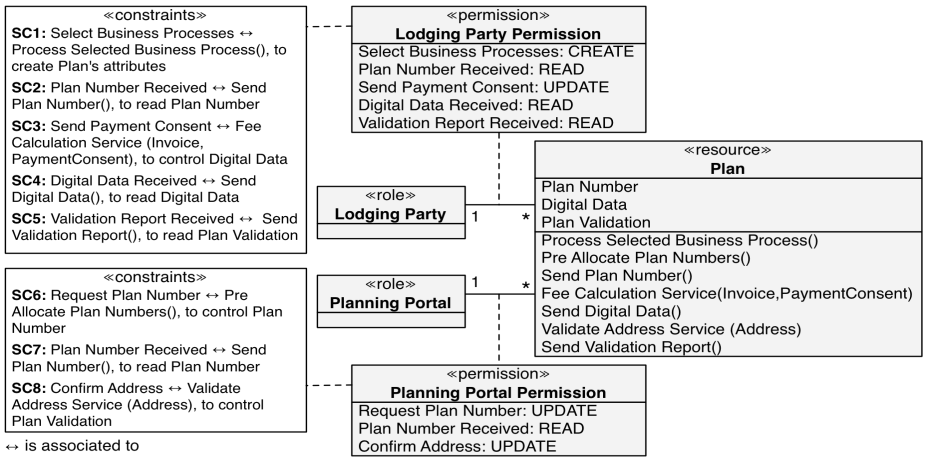


Fig. 4: RBAC Security Model - Prepare Plan Business Process

RQ1. Lodging Party should be able to:
1. create or initialize the Plan Number, Digital Data and Plan Validation.
2. read the Plan Number, Digital Data and Plan Validation.
3. update the Digital Data.

RQ2. Planning Portal should be able to:
1. update the Plan Number and Plan Validation.
2. read the Plan Number.

The security model (i.e., Fig. 4) defines how authorised parties should access the protected resources. However, it does not support capturing the concerns related to the separation of duties, binding of duties, and usage control [1]. The security requirements RQ3, RQ4 and RQ5 should be taken into consideration:

RQ3. Secured operations (e.g., Fee Calculation Service) should be performed by different users assigned to the same role.
RQ4. A sequence of secured operations (e.g., Pre allocate Plan Numbers and Send Plan Number) should be performed by the same user assigned to the role (e.g., Planning Portal).
RQ5. The system (i.e., Registry) should place constraints on how confidential data should be used by the roles (i.e., Lodging Party and Planning Portal).

RQ3 defines that there should exist at least two users in the Registry with the same role, to finish executing the task Fee Calculation Service: the first user issues the Invoice and the second user approves the Payment Consent. Requirement RQ4 highlights the concept binding of duties. Requirement RQ5 defines the security constraints for usage control; e.g., the Registry could potentially define constraints for Digital Data and Validation Report saying, that they remain valid for seven days. Elicitation of requirements RQ3-5 much depends on the concrete problem. They can't be captured from the business model and require involvement of business and/or security analysts.

Communication Channel is used to exchange data between business partners (e.g., Lodging Party and Planning Portal) and system (e.g., Registry). Here, data, like Selected Business Process(es), Payment Consent and etc, need to be protected when they are transmitted over the (untrusted) communication channel, i.e., Internet. The communication channel could be intercepted by the threat agent and the captured data could be misused (i.e., read and kept for the later use or modified and passed over) by the threat agent. This could lead to the loss of the channel reliability, and could negate the confidentiality and integrity of the Plan. To mitigate the risk, the requirements should be implemented for the Lodging Party and Registry and correspondingly for the Planning Portal and Registry:

RQ6. The server (e.g., Registry) should have the unique identity in the form of key pairs (public key, private key) certified by a certification authority.
RQ7. The client (e.g., Lodging Party and Planning Portal) should encrypt and sign the data (e.g., Selected Process(es), Plan Number, and other) using keys before sending it to the server (e.g., Registry).

A security requirements implementation could be fulfilled by the standard transport layer security (a.k.a., TLS) protocol [3] as illustrated in Fig. 5. As the first contact, the Lodging Party sends Registry a handshake message, which includes a random number. Following RQ6, the Registry responds with its public key and the information about the certification authority. After verification of the Registry's public key, the Lodging Party generates the secret and sends it to the Registry encrypted with the Registry's public key. The Registry then decrypt the secret using the private key and generates symmetric session keys. The keys enable Lodging party and Registry to establish a secure session for data exchange.

Following RQ7, encryption keeps the transmitted data (e.g., Selected Business Process(es), Payment Consent and etc) confidential and signing it ensures that the received data is not tempered. The secure communication continues until it is not explicitly terminated by Lodging Party or Registry.

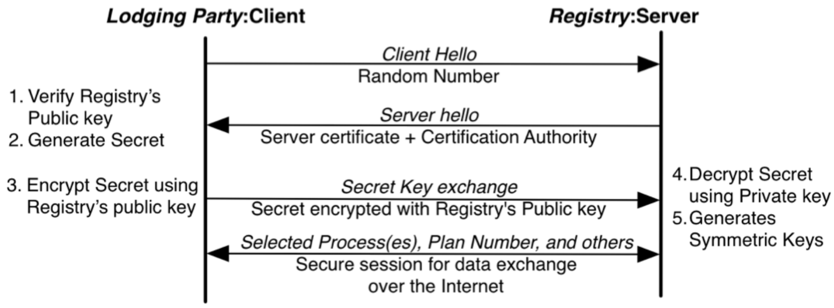


Fig. 5: TLS Protocol implementation, adapted from [3]

Input interfaces are used to input data submitted by business partners. In Fig. 2, we identify Process Selected Business Process(es) and Fee Calculation Service as input interfaces of Registry that receives the Selected Process(es) and Payment Consent from Lodging Party. The threat agent can exploit the vulnerability of the input interfaces by submitting the data with a malicious scripts. If happening so the availability and integrity of any activity (e.g., Send Digital Data) after the input interface (e.g., Fee Calculation Service) may be negated. To avoid this risk the following security requirements must be implemented for the input interface:

RQ8. The input interface (e.g., Fee Calculation Service) should filter the input data (e.g., Payment Consent).

RQ9. The input interface (e.g., Fee Calculation Service) should sanitize the input data (e.g., Payment Consent) to transform it to the required format.

RQ10. The input interface (e.g., Fee Calculation Service) should canonicalize the input data (e.g., Payment Consent) to verify against its canonical representation.

Input filtration [5] (RQ8) validates the input data against the secure and correct syntax. The string input should potentially be checked for length and character set validity (e.g., allowed and blacklisted characters). The numerical input should be validated against their upper and lower value boundaries. *Input sanitization* (RQ9) should check for common encoding methods used (e.g., HTML entity encoding, URL encoding, etc). The *input canonicalization* [5] (RQ10) verifies the input against its canonical representation.

Business Service is a task or activity executed within an enterprise on behalf of the business partner [6]. The goal is to guarantee availability of the business services. The business services, like Fee Calculation Service offered to Lodging Party, are provided by the server (e.g., Registry) through the communication channel. The threat agent may exploit the hosts in the channel and hack them because of the protocol (e.g., TCP, ICMP or DNS [4]) vulnerability; i.e., the ability to handle an unlimited number of requests for service. When receiving simultaneously multiple requests, the server i.e., Registry, will not be able to handle them, thus, the services become unavailable. The successful denial of service attacks could also provoke the loss of partner's (e.g., Lodging Party and Planning Portal) confidence on Registry. To mitigate this risk, one could define three types of firewalls (see Fig. 6) – Packet Filter Firewall, Proxy Based Firewall and Stateful Firewall [12], and introduce the following requirements:

RQ11. Server (e.g., Registry) should establish a rule base (i.e., a collection of enterprise' constraints used by different firewalls) to communicate with the business partners (e.g., Planning Portal).
RQ12. Packet Filter Firewall should filter the business party's (e.g., Planning Portal's) address to determine if it is not a host used by the threat agent.
RQ13. Proxy Based Firewall should communicate to the proxy which represents the business service (e.g., Pre allocate Plan Number) to determine the validity of the request received from the business party (e.g., Planning Portal).
RQ14. State Firewall should maintain the state table to check the party's (e.g., Planning Portal's) request for additional conditions of established communication.

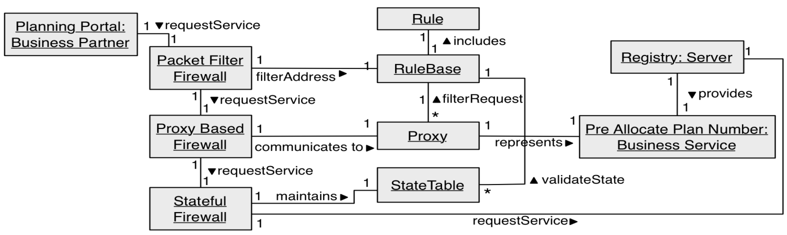


Fig. 6: Firewall Architecture, adapted from [12]

It is important to notice that the communication between the Planning Portal (and also Lodging Party) and the Registry is bidirectional. The similar requirements must be taken into account when Registry sends messages (e.g., Fee Calculation Service sends Invoice) back to the business party.

Data Store is used to define how data are stored and retrieved to/from the associated databases (e.g., Data store in Fig 2). If the threat agent is capable of accessing and retrieving the data, their confidentiality and integrity would potentially be negated, thus, resulting in the harm of the business asset (i.e., the Plan) and its supporting IS assets (i.e., the Registry).
RQ15. The server (e.g., Registry) should audit the operations after the retrieval, storage or any other manipulation of data in the data store (e.g., Data store).

Auditing is the process of monitoring and recording selected events and activities [9]. It determines who performed what operations on what data and when. This is useful to detect and trace security violations performed on the Plan Number, Digital Data and Plan Validation. Potentially, the data store auditing could be supported by the access control policy.
RQ16. The server (e.g., Registry) should perform operations to hide/unhide data when they are stored/retrieved to/from the data store (e.g., Data store).

A possible RQ16 implementation is cryptographic algorithms. The encryption offers two-fold benefits: (*i*) the data would not be seen by the Data store users (e.g., database administrator) where the circumstances do not allow one to revoke their permissions; (*ii*) due to any reasons if someone gets physical access to the Data store (s)he would not be able to see the confidential data stored.

4 Conclusion

In this paper, we presented a method to elicit security requirements from the business process models. Its strength lies in its general description of security goals and the systematic analysis of the contextual areas. In comparison to the related work where the focus is placed on representing security requirements (graphically) on the process models, our proposal suggests a novel approach to elicit these requirements and define them as the business rules.

The method should be improved with new security risk-oriented patterns. We also plan to extent the method with the requirements prioritisation to support the trade-off analysis. Finally, we continue validating the method empirically.

References

1. Accorsi, R., Stocker, T.: On the Exploitation of Process Mining for Security Audits: The Conformance Checking Case. In: Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 1709–1716. SAC, ACM (2012)
2. Ahmed, N., Matulevičius, R.: Securing Business Processes using Security Risk-oriented Patterns. Computer Standards and Interfaces 36(4), 723–733 (2014)
3. Apostolopoulos, G., Peris, V., Saha, D.: Transport Layer Security: How Much Does It Really Cost? In: Proceedings IEEE INFOCOM'99 The Conference on Computer Communications. vol. 2, pp. 717–725 (1999)
4. Chang, R.: Defending Against Flooding-based Distributed Denial-of-Service Attacks: A Tutorial. Communications Magazine, IEEE 40(10), 42–51 (2002)
5. Clarke, J., Fowler, K., Oftedal, E., Alvarez, R.M., Hartley, D., Kornbrust, A., O'Leary-Steele, G., Revelli, A., Siddharth, S., Slaviero, M.: SQL Injection Attacks and Defense, Second Edition. Syngress Publishing, 2nd edn. (2012)
6. Dumas, M., O'Sullivan, J., Hervizadeh, M., Edmond, D., Hofstede, A.H.M.t.: Towards a Semantic Framework for Service Description. In: Semantic Issues in E-Commerce Systems. pp. 277–291. Kluwer, B.V. (2003)
7. Herrmann, P., Herrmann, G.: Security Requirement Analysis of Business Processes. Electronic Commerce Research 6(3-4), 305–335 (2006)
8. Menzel, M., Thomas, I., Meinel, C.: Security Requirements Specification in Service-Oriented Business Process Management. In: ARES. pp. 41–48 (2009)
9. Natan, R.B.: Implementing Database Security and Auditing: Includes Examples for Oracle, SQL Server, DB2 UDB, Sybase. Digital Press, Newton, MA, USA (2005)
10. Rodríguez, A., Fernández M, E., Piattini, M.: A BPMN Extension for the Modeling of Security Requirements in Business Processes. IEICE-TIS(4) pp. 745–752 (2007)
11. Röhrig, S., Knorr, K.: Security Analysis of Electronic Business Processes. Electronic Commerce Research 4(1-2), 59–81 (2004)
12. Schumacher, M., Fernandez B., E., Hybertson, D., Buschmann, F., Sommerlad, P.: Security Patterns: Integrating Security and Systems Engineering. Wiley (2006)

Automated Adaptation of Business Process Models Through Model Transformations Specifying Business Rules

Roman Popp and Hermann Kaindl

Vienna University of Technology, Vienna, Austria
{roman.popp, hermann.kaindl}@tuwien.ac.at

Abstract. Both business processes and business rules are changing over time. In addition, when business process models try to capture all details, they are complex and need to deal with variability according to context. Manual adaptation of process models takes time and is error-prone.
We address these issues through explicit separation of process models (represented in BPMN) and specifications of business rules (represented as model-transformation rules based on BPMN). Applying such rules as model transformations to a model of a reference process (at design-time) leads to models of either refined or new models of business processes. In this way, both the change of a process per se or the inclusion of a new business rule can be handled through automated adaptation of the related business models, without an extension of the BPMN 2.0 standard.

Keywords: Business process adaptation, Model-driven transformation, BPMN

1 Introduction

Business processes in organizations are changing over time, so their models need to be kept up-to-date. Usually, such models are represented today in Business Process Model and Notation (BPMN), more precisely its version BPMN 2.0, since this language offers a standard for modeling business processes.[1]

Business processes also need to take variability into account, depending on certain conditions. According to, e.g., Hallerbach and Bauer [1], neither conditional branches inside a process nor separate process models are satisfactory solutions for representing process variability, since both can result in redundancies.

Business rules often define such variability, but informally. Manual adaptations of business processes according to business rules are tedious and lead to such redundancies.

Therefore, research approaches (discussed below) propose automated adaptation of reference processes. We present a new approach in this paper based on model-transformation, which is a core technology of the Object Management Group's Model Driven Architecture[2] (MDA). Model-transformation employs rules that transform a

[1] http://www.omg.org/spec/BPMN/2.0/
[2] http://www.omg.org/mda/

given source model to a related target model, according to precisely specified metamodels.

Our new idea for automated adaptation of business processes presented in this paper is to represent certain kinds of business rules as model-transformation rules. When specified precisely in this way, they can be more or less automatically used to adapt business processes represented as models. For BPMN 2.0, a corresponding metamodel exists and is used by our approach. Since we transform from BPMN to BPMN, the same metamodel defines both source and target model in our approach.

We present our new approach with a running example of a business rule in the context of payment and its conditional authorization. We also show how BMPN models can actually be modified automatically through such model transformations.

The remainder of this paper is organized in the following manner. First, we discuss related work. Then we explain specifying a business rule as a model-transformation rule. Based on that, we explain automatically adapting reference processes through model transformations. Finally, we discuss our approach more generally, draw some conclusions and indicate future work.

2 Related Work

In product-line engineering of software or general systems, the core theme is commonality and variability, see e.g., [2] for an approach and its formalization in the context of requirements and systems engineering. Most of this work is based on feature-oriented models as introduced early by Kang et al. [3], which typically distinguish between mandatory and optional features, as well as mutual exclusion and lists of choices. Mannion and Kaindl [2] integrated parameters into such models as well. A product line in these approaches usually encompasses all (pre-)defined variants. We are not aware, however, of any application of these ideas to adaptability of business process models.

In this paper, we deal with business processes modeled in standard BPMN 2.0, and business rules for their adaptation. So, we also focus the further discussion of related work on this topic. Still, let us also mention an earlier approach to achieving business process flexibility with business rules [4]. However, the points of variability in the reference process have to be identified manually.

Hallerbach and Bauer [1] pursue an approach named Provop, where a business process variant constitutes an adjustment of a *reference process*. Either a standard process or the most frequently used process can be taken as the reference process. Provop defines specific change operations (with pre- and post-conditions) for adaptation of the reference process, which may also be grouped into so-called options. It is not so clear, however, how business rules for adapting a reference process can, in general, be formulated in such an approach, so that they are understandable per se.

Milanovic et al. [5] devised a dedicated language for rule-based sub-process selection and workflow composition called rBPMN. It extends BPMN through weaving with the R2ML rule language. Since the resulting models are, in general, not compatible with the BPMN standard, however, the usual tooling for BPMN cannot be directly used with this approach.

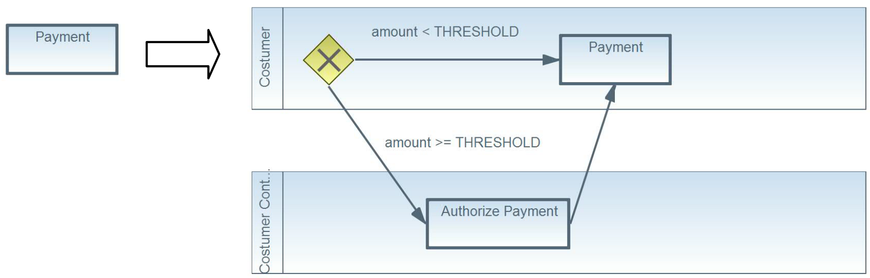


Fig. 1. Business rule example with in-situ substitution

La Rosa and Dumas [6] propose modeling one large reference workflow. Their approach provides design-time configuration for realizing workflow variants, but there is no extensible approach for including business rules.

Döhring et al. [7] defined a new metamodel for so-called vBPMN models resulting from *weaving* BMPN2 and business rules (specified in the R2ML rule language). This approach allows specifying change patterns for application at run-time. However, the reference workflow requires to have adaptive segments explicitly marked. Like rBPMN, vBPMN is an extension of BPMN (as standardized), so that the usual tooling for BPMN cannot be directly used with this approach.

Döhring et al. [8] address this latter problem through design-time model transformations from vBPMN to BPMN2 Sub-Processes (as standardized), where the transformations are hard-coded in Java. Still, the reference model requires adaptive segments to be marked.

In summary, we are not aware of any approach that allows modeling and executing business rules specified directly as model transformations according to MDA, where both source and target model conform to the BPMN 2.0 metamodel.

3 Business Rules Specified as Model Transformation Rules

In our approach, we specify business rules as model transformation rules, as independently as possible from any existing process model as possible. Of course, such a business rule has to refer to something in some existing business process, in order to be useful at all. However, these rule specifications and the process specifications should be maximally decoupled. So, we explain simple examples of such business rules even before referring to any concrete reference process.

Our chosen example deals with payment, a very usual part of typical business processes. Let us assume that payment is to be handled differently in a given business depending on the amount to be paid. If this amount exceeds a defined threshold, another business actor than the one primarily responsible for executing a payment Activity needs to authorize the payment before its execution.

One possibility for specifying this business rule in our approach is illustrated in Fig. 1, which implements an “in-situ” substitution (at the same place) of a given Payment Activity with the conditional payment authorization. In addition to this graphical illustration, such a rule needs to be specified in more detail for making it automatically executable. We specified this rule as a transformation in the ATLAS Transformation Language (ATL).[3] Each ATL rule consists at least of a *from* and a *to* part. The *from* part specifies which element is transformed (possibly with some conditions) and the *to* part specifies the corresponding part in the target model. These parts are based on BPMN in our approach, more precisely BPMN 2.0, where all the technical details have been specified in an XML representation.

We also specified this business rule in different ways, but due to lack of space, we cannot explain them all here. Still, let us mention another formalization for simply transforming a Payment Activity into a BPMN model through a Payment Sub-Process that includes additionally an authorization Activity and conditional Payment. This transformation also needs to create this Sub-Process in BPMN in such a way, that it contains this conditional payment processing.

4 Automatically Adapting Reference Processes through Model Transformations

Based on such specifications of business rules, our approach can automatically adapt reference processes through model transformations. A small example of a reference process represented in BPMN is shown in Fig. 2. It is a simplified version of the "payment handling" process of [9, p. 108]. This payment handling process starts with the creation of an invoice. This invoice is then sent to the customer. After the customer has received the invoice, she makes the payment of the invoice. Concurrently, the delivering company checks the receipt of payment (through its bank). Until the payment is received, the check is repeated periodically. Once the payment is received, it is booked as paid. After that, this example reference process is finished. Note, that payment is unconditional in this process.

This process specification per se does not include any variability. Of course, it could be simply changed manually to include the conditional payment authorization as desired in the business rule example above. This would encode this particular rule in this particular process model, with all the disadvantages already discussed in the literature (see also above).

Much as other approaches, we strive for automated adaptation, but we pursue it through model transformation. After the given business rule as shown in Fig. 1 had been formalized in ATL, applying the ATL rule engine resulted in a process model as shown in Fig. 3. This rule replaces the simple Payment Activity in a reference process with a whole process part directly inside the adapted reference process. In fact, we had to implement several technical work-arounds to make this work because of restrictions of this transformation approach. Due to lack of space, we cannot specify these here.

5 Discussion

It may seem as though it is straight-forward to add more and more business rules as transformation rules according to our approach. However, certain constraints have to

[3] `http://www.eclipse.org/atl/`

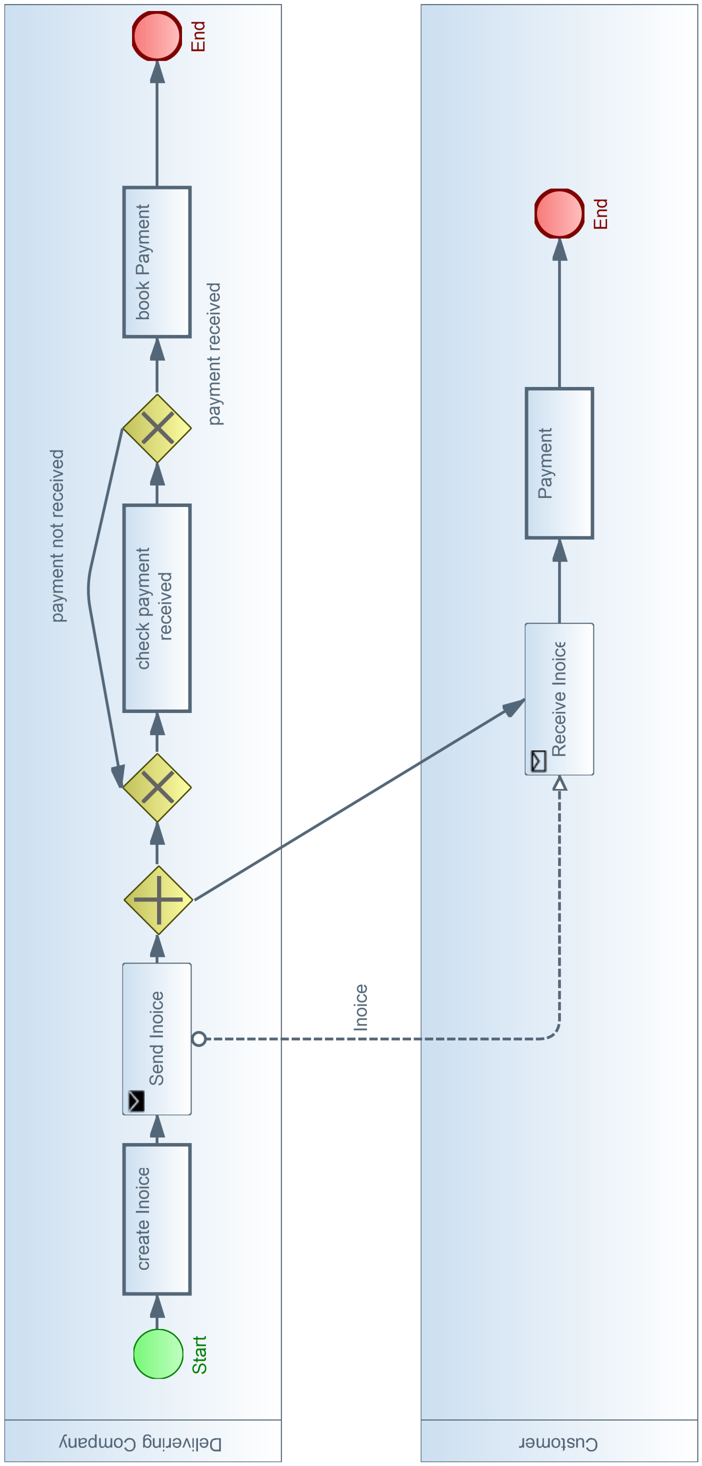


Fig. 2. Reference Process

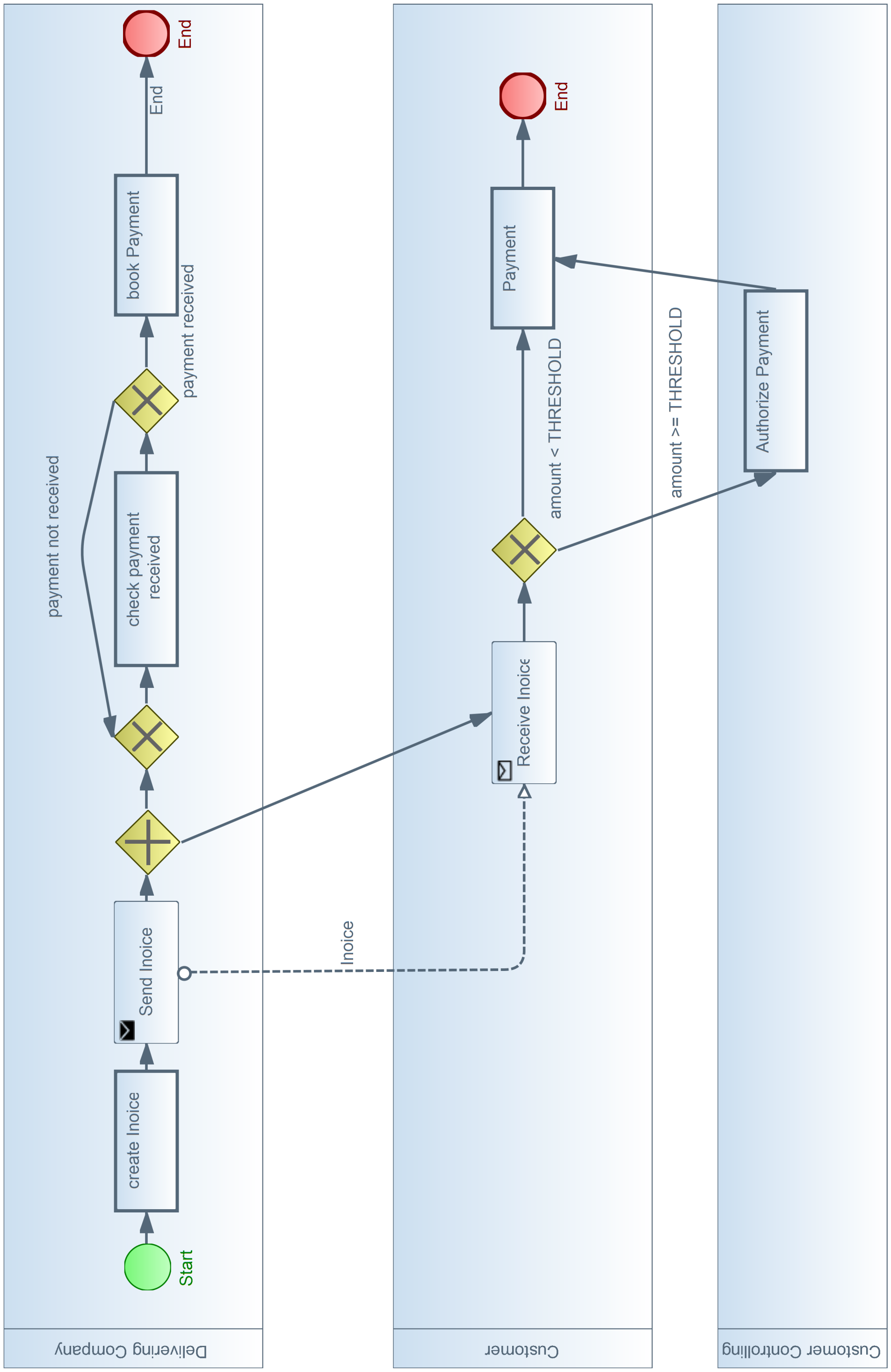


Fig. 3. in-situ Changed Process

be fulfilled, so that the business process resulting through model transformation still conforms to BPMN. For example, it is necessary that the *from* and *to* parts have the same input and output "ports". In our simple example, these are one entry point and one exit point. These constraints will have to be studied yet, possibly along the lines of [10].

Our approach presented in this paper allows strong decoupling of (new) business rules from existing business processes. So almost any existing process can potentially be adapted according to new business rules automatically.

It is also an advantage of our approach that no segments of existing processes need to be marked for adaptation, and so the designer does not need to think of possible adaptions at all, because all the adaptions are fully specified in the business rules.

We only want to modify parts of the models, but the model transformation engines are made for transforming a whole model. This leads to large overhead, like providing generic transformation rules, and because of their triggering restriction we also need the approach of Wagelaar et al. [11] for circumventing this technical problem.

A more convenient solution could be based on a specific transformation engine providing triggering and firing of more than one rule for the same source element. It could be devised along the lines of an existing one as specifically implemented in another context [12, 13], which makes the approach of Wagelaar et al. unnecessary.

6 Conclusion and Future Work

The work presented in this paper addresses adaptability of business process models and, in particular, automating it. We follow the approach to separate reference processes from business rules and propose here the new idea to represent certain kinds of business rules as model transformations.

Our implementation assumes reference processes to be represented as standard BPMN 2.0 models. The *from* and *to* parts of each transformation rule are based on BPMN 2.0 as well, so the process models resulting from such transformations are also standard BPMN. Therefore, existing BPMN execution engines / environments can be used for the resulting process models.

Viewed from a higher perspective, this approach leads to a new form of representing and managing variability of business process models. When specifying reference processes, it is not yet necessary to envisage their variability. In particular, no marking of variable parts is necessary. Vice versa, when specifying business rules in this way, the available reference processes need not be taken into account as a whole or in detail. Just knowing about parts of such processes or sub-processes is required for specifying the business rules.

Still, future work will be necessary for studies with people actually working on adaptable business processes. In order to avoid the technicalities when using currently available transformation engines, creating an advanced one will facilitate the adoption of this approach. Such an engine will not necessarily have to be general enough to handle any metamodel, but support specifically the one of BPMN 2.0. The use of such a (specific) transformation engine will support the specification of business rules as transformation rules.

7 Acknowledgments

Part of this research has been carried out in the ProREUSE project (No. 834167), funded by the Austrian FFG.

References

1. Hallerbach, A., Bauer, T., Reichert, M.: Configuration and management of process variants. In: Intl. Handbook on BPM. Springer Berlin / Heidelberg (2010) 237–255
2. Mannion, M., Kaindl, H.: Using parameters and discriminants for product line requirements. Systems Engineering **11**(1) (2008) 61–80
3. Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, A.: Feature-Oriented Domain Analysis (FODA) feasibility study. Technical report, Carnegie Mellon University, Pittsburgh, PA (November 1990)
4. van Eijndhoven, T., Iacob, M.E., Ponisio, M.: Achieving business process flexibility with business rules. In: Proceedings of the 12th Enterprise Distributed Object Computing Conference (EDOC'08). (Sept 2008) 95–104
5. Milanovic, M., Gasevic, D., Rocha, L.: Modeling flexible business processes with business rule patterns. In: Proceedings of the 15th Enterprise Distributed Object Computing Conference (EDOC'11). (2011) 65–74
6. La Rosa, M., Dumas, M., t. Hofstede, A., Mendling, J.: Configurable multi-perspective business process models. IS **36**(2) (2011)
7. Döhring, M., Zimmermann, B.: vBPMN: Event-Aware Workflow Variants by Weaving BPMN2 and Business Rules. In: Proceedings of the 16th International Conference on Exploring Modeling Methods in Systems Analysis and Design. Volume 81 of LNBIP., Springer (2011) 332–341
8. Döhring, M., Schulz, A., Galkin, I.: Emulating Runtime Workflow Adaptation and Aspect Weaving by Recursive Rule-Based Sub-Process Selection - A Model Transformation Approach. In: Proceedings of the 16th Enterprise Distributed Object Computing Conference (EDOC'12), Beijing, IEEE (2012)
9. Schumacher, J., Meyer, M.: Customer Relationship Management strukturiert dargestellt: Prozesse, Systeme, Technologien. Springer-Verlag (2003)
10. Döhring, M., Heublein, S.: Anomalies in Rule-Adapted Workflows - A Taxonomy and Solutions for vBPMN. In: Proceedings of the 16th European Conference on Software Maintenance and Reengineering, Szeged, Hungary, IEEE (2012) 117–126
11. Wagelaar, D., Van Der Straeten, R., Deridder, D.: Module superimposition: a composition technique for rule-based model transformation languages. Software and Systems Modeling **9** (2010) 285–309 10.1007/s10270-009-0134-3.
12. Popp, R., Raneburger, D., Kaindl, H.: Tool support for automated multi-device GUI generation from discourse-based communication models. In: Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '13), New York, NY, USA, ACM (2013)
13. Raneburger, D., Popp, R., Kaindl, H.: Model-driven transformation for optimizing PSMs: A case study of rule design for multi-device GUI generation. In: Proceedings of the 8th International Joint Conference on Software Technologies (ICSOFT'13), SciTePress (July 2013)

Managing Quality Related Information in Software Development Processes

Vladimir A. Shekhovtsov, Heinrich C. Mayr

Institute for Applied Informatics, Alpen-Adria-Universität Klagenfurt, Austria
{Volodymyr.Shekhovtsov,Heinrich.Mayr}@aau.at

Abstract. An effective communication between the parties in the software development process is important for coming to and complying with appropriate agreements on the quality of the prospective software. Such communication is impaired when developers and business stakeholders perceive quality differently. To address this problem, we aim at a solution that supports understandability and reusability of quality-related communicated information, and the quality of decisions based on this information. In this paper, we first introduce a set of knowledge structures for representing communicated information and then discuss how to map raw communication data into these structures.

Keywords: elicitation, semantic annotation, software development process, software quality, communicated information

1 Introduction

Software development processes require a continuous involvement of the affected business stakeholders in order to be successful (this requirement, in particular, is reflected by the ISO/IEC standard for software life cycle processes [6]). A prerequisite of such involvement is establishing an appropriate communication basis for the different parties. In particular, such a basis is needed for coming to terms and agreements on the quality of the software under development. Without this, quality defects are often detected only when the software is made available for acceptance testing.

Clearly, besides of enabling effective communication, the communicated quality-related information has to be managed properly and made available during the software development lifecycle; moreover, as past-experience may help to take the right decisions, such information should be provided in a way that allows for easy access (e.g., via an issue management system) and analysis.

The QuASE project[1] [11] aims at a comprehensive solution for these issues. In particular, this solution will provide support for managing (1) the **understandability** of quality-related *communicated information*, (2) the **reusability** of that information, and (3) the **quality of decisions** based on that information.

---

[1] QuASE: Quality Aware Software Development is a project sponsored by the Austrian Research Promotion Agency (FFG) in the framework of the Bridge 1 program (http://www.ffg.at/bridge1); Project ID: 3215531

In this paper, we concentrate on the knowledge structures representing quality-related communicated information and on the mapping of raw communication data into these knowledge structures.

The paper is structured as follows. Section 2 introduces the sources of quality related information and defines the knowledge structures representing QuASE quality characteristics. Section 3 describes the mapping of communicated information into these knowledge structures. After a short discussion of related work in Section 4, the paper concludes with a summary and an outlook on future research (Section 5).

2 Knowledge Structures for Representing Quality Related Communicated Information

Usually, industrial software development projects keep communicated information within *repositories* such as

1. *project databases* controlled by issue management systems, e.g., JIRA [7], MantisBT [6] and others; such databases contain communicated information in form of so-called issues that (generalizing communication units as bug reports or feature requests) and related discussions;
2. *file-based repositories* containing meeting minutes, requirement and design specifications etc.; these files are usually kept in some kind of a directory tree under the control of configuration management systems; these documents, as a rule, are updated less frequently as compared to issues and the relevant discussion threads;
3. *wiki-based systems*.

Consequently, QuASE considers these types of repositories as sources of quality-related communicated information and therefore provides interfaces to them.

The raw data collected from these sources are interpreted and mapped into the QuASE *QuIRepository* the structure of which (defining a generic metamodel representing semantic relationships) is depicted in figure 1 and subsequently explained.

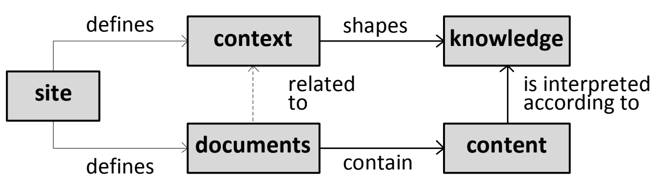


Fig. 1. Generic structure of QuIRepository

1. ***QuASE site:*** owner of the given QuASE installation, e.g. a software provider.
2. ***QuASE context:*** units having particular views on communicated information, e.g. projects, organizations and their departments, involved people (stakeholders) etc.. Context units are characterized by context attributes and can be connected to other units; a context configuration, for example, could include the representation of the

whole organizational hierarchy or the whole portfolio of projects defined for a particular IT company.

3. ***QuASE documents:*** units shaping communicated information: they serve as containers for such information or organize such containers. We distinguish *content holders* and *content directories* organizing the holders. Examples of document units are issues and their sets, issue attribute values, requirement specifications and their structural elements. For the case of issues, the issues or their sets are examples of content directories, whereas issue descriptions and discussion opinions are examples of content holders. Document units can be related to particular context element. A detailed description of the context and document concepts is target of a separate publication.
4. ***QuASE knowledge:*** quality and domain knowledge that is subject of communication and harmonization. We organize it into knowledge modules representing particular views. The configuration of these modules reflects the configuration of context, i.e., the modules and their relationships correspond to context elements and their interrelationships. Below, these modules will be described in more detail.
5. ***QuASE content:*** the information that has to be communicated. It is shaped by context units and interpreted according to the respective knowledge. Dealing with the content is decoupled from dealing with their holder documents; i.e., we can think of this content as of a uniform stream of data (which is given as tagged natural text in the current QuASE implementation). On the other hand, while dealing with documents, we abstract from their content and delegate dealing with this content to the generic content processing routines.

The QuASE knowledge modules are organized into a modular ontology (QuOntology) thus providing a framework for translating between world views. Initial research on QuOntology has been published in [13], whereas the current version of the relevant conceptualizations is presented in [11].

QuOntology is organized in three layers [see also Fig. 2]:

1. ***QuOntology core*** represents a stable subset of the knowledge available from research and industrial practice; this knowledge does not depend on the particular problem domain and the particular context. We use the Unified Foundational Ontology (UFO) [4, 5] as a foundation for QuOntology core.
2. ***Domain ontologies*** [5] represent the specifics of the particular problem domain which is addressed by the given software under development (finance, oil and gas etc.); domain ontology concepts specialize core concepts; as a part of the project, we implement for this layer an ontology for quality in the software domain [11].
3. ***Context ontologies*** represent the knowledge related to particular components of the QuASE context: they contain organization-specific, project-specific etc. concepts. These concepts specialize the generic concepts of the upper level ontologies but also may be specializations of other context ontologies; we implement for this layer ontologies for business-specific and IT-specific views on quality;

To deal with changes in the structure of context and document units, we will provide a notation for specifying context and document configurations, which will be supported by the metamodeling tool ADOxx (http://www.adoxx.org); this will allow the responsible people (knowledge suppliers) to create and modify the desired configuration. The relevant database structure will be generated based on this configuration.

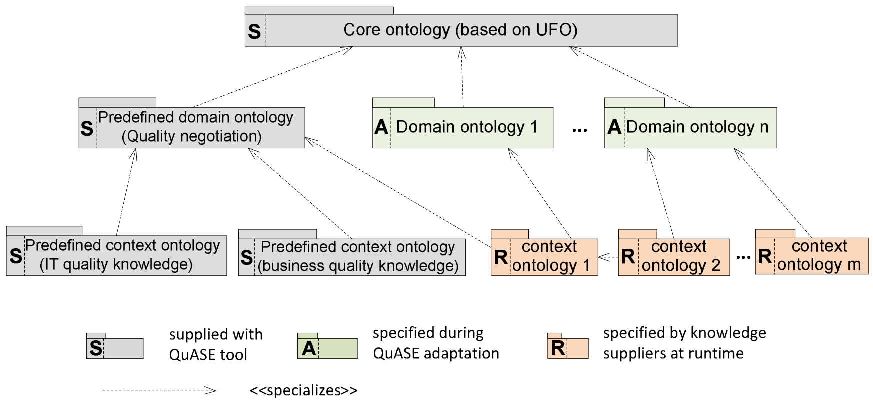


Fig. 2. QuASE ontology layers (adapted from [12])

3 Mapping communicated data into QuIRepository structures

The correspondence of communicated data and QuIRepository structures is made explicit by *mapping specifications* of a particular *mapping mode*.

For brevity, in this section we restrict ourselves to the mapping of repositories that are controlled by an issue management system (e.g. JIRA databases) further referred to as *mapping sources*. Mappings involving other categories of repositories (e.g. file-based repositories or wikis) are based on similar principles. Also, we omit the treatment of the mapping of concept relationships.

3.1 Mapping context structure

To define appropriate **mappings for the context concepts** we distinguish the following mapping modes:

1. ***Direct mode***: a given communicated context-related structure (e.g. a JIRA database table) is mapped one-to-one into a QuIRepository context structure;
2. ***Join mode***: several communicated context-related structures are mapped into a single QuIRepository context structure;
3. ***Split mode***: a single communicated context-related structure is mapped into several QuIRepository structures;
4. ***Interactive mode***: the whole instance of the context concept has to be specified by the user through the respective user interface.

Specifications for **mapping context attributes** are nested into the specifications defined for context concepts. We distinguish the following mapping modes:

1. ***Direct mode:*** a single communicated attribute (e.g. defined as an attribute in a context-related relation) is mapped into a single QuIRepository context attribute. The data is extracted without any user interaction. Example: mapping the "project name" attribute of a JIRA project table to the "project name" attribute of the corresponding QuIRepository "project" context unit;
2. ***Calculated mode:*** one or several communicated attributes are mapped to a QuIRepository context attribute based on a predefined metric function;
3. ***Interactive mode:*** the QuIRepository context attribute cannot be derived automatically from the communicated data; in this case, the QuASE tool shows an elicitation user interface and collects the concept information from the expert user.

3.2 Mapping document structures

1. ***Mapping document concepts:*** is defined similarly to the direct context mapping mode: the communicated document structure is mapped to a specific QuIRepository document structure. As an example, a JIRA "issue" table is mapped to the "issue" document structure, whereas the comments to the issues or issue descriptions are mapped into, correspondingly, "comment" or "issue description" content holders.
2. ***Mapping document attributes***: For the document attributes, the mapping is defined through the same three modes as specified above for context attributes; the main difference is due to the fact that it is possible to distinguish calculated attributes based on the content held directly by the document unit (for the case of content holders) or by the related holders (for the case of content directories).
3. ***Mapping content holders:*** For content holders, the approach is to delegate all the processing of mapping the content to the specific content-mapping activities such as text-based semantic annotation as specified in the following section.

3.3 Mapping content

Mapping content stream into QuASE concepts is performed by associating concepts with text fragments of the stream. To perform such an association, the QuASE system:

1. scans the natural language content stream looking for candidate context-specific terms;
2. associates tags with candidate terms that correspond to available knowledge in QuOntology; applying a tag indicates that the corresponding term can be associated with a QuOntology concept in at least one ontology module;
3. makes the tags act as anchors for connections to the related QuOntology concepts; to do this, for every tagged term the tool looks for concepts in all available context ontology modules.

The *Term knowledge context* then is the set of all concepts found for the given term; it defines all possible context-specific views of this term, and allows for switching between such views.

Fig.3 visualizes the process of associating context-specific terms with tagged documents exemplified by JIRA issues.

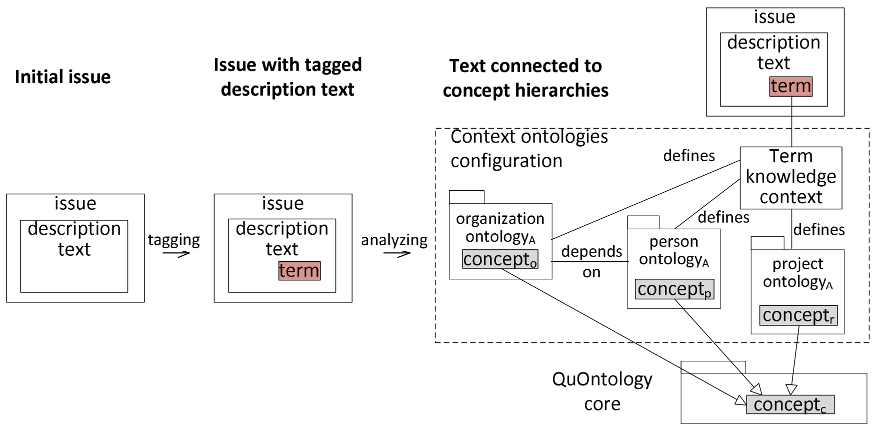


Fig. 3. Associating context-specific terms with tagged JIRA issues (domain ontology layer is omitted)

4 Related work

In this section, we discuss two categories of the related work: (1) approaches addressing the complete set of goals for QuASE, and (2) approaches addressing the particular task of obtaining the data from project repositories for analytical purposes.

4.1 Knowledge and experience management solutions

The approach discussed here belongs to the category of solutions that facilitate storing, reusing, adapting, and analyzing the development knowledge. In particular such solutions apply the existing body of research on knowledge management to the field of software engineering [1, 2]. A more specific category of solutions is related to managing past software engineering experience; they are known as experience management solutions [10].

With respect to our aims, these solutions bear the following shortcomings: (1) they do not specifically address quality-related issues, which is true especially for those issues that are available from existing repositories like issue management systems; (2) they collect the experience only as viewed from the developer side; the business stakeholder's view is mostly ignored, and it is not possible to switch between views while considering collected experience.

4.2 Solutions for obtaining information from project repositories

Approaches that aim at obtaining information from project repositories for analytical purposes, typically belong to the research area of *mining software repositories* [8]. Particular examples of such approaches include automatic categorization of defects [14], building software fault prediction models based on repository data [15], and using repositories to reveal traceability links [9]. Other approaches use repository information to analyze the applicability of specific development practices [3].

Repository mining solutions use software repositories as sources of quantitative code- and coding process-related information (such as the frequency of bugs, the time spent on various tasks, information about commits into repositories etc.). In contrast to that, QuASE uses repositories as sources for communicated information by looking into issue descriptions, negotiation opinions, wikis, and requirements documents.

In addition to the difference in the general goals, the QuASE approach differs from these solutions in the following implementation-related aspects: (1) it conceptualizes the process of collecting information from repositories as mapping operations controlled by mapping specifications; (2) it is based on an established set of conceptual structures that represent context and document units, content stream, and view-specific ontological knowledge.

5 Conclusions and future work

In this paper, we outlined a solution that is intended to support understandability and reusability of quality-related information, and thus may help to improve the quality of decisions in the software development process. The QuASE provides a knowledge-oriented interface to information that is communicated and collected in the course of software development projects. For this purpose, we introduced a set of knowledge structures addressing quality characteristics; these include context-, document-and content-specific structures as well as the structures for knowledge that defines particular views. We then defined the various kinds of modes for mapping communicated information (such as the data available in the project databases or document repositories) into these knowledge structures.

Ongoing research within the framework of the QuASE project aims at realizing the following features based on the defined conceptual structures:

1. *Understandability support:* document units are analyzed with respect to potential understandability problems for the target context (e.g., when they units are to be presented to a non-expert business stakeholder; identified problems are solved by translating or explaining the problematic terms using the respective knowledge structures.
2. *Reusability support:* for a given document, all similar ones (with respect to the required knowledge level) are searched based on the attributes of the documents and/or context units.
3. *Quality of decisions support:* recommendations for dealing with documents and context elements during the communication; forecasts of metrics values, and performing “what-if” analyses for particular decisions.

References

1. Aurum, A., Jeffery, R., Wohlin, C., Handzic, M. (eds.): Managing Software Engineering Knowledge. Springer, Berlin-Heidelberg (2003)
2. Bjørnson, F.O., Dingsøyr, T.: Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and Software Technology 50, 1055-1068 (2008)
3. Ernst, N.A., Murphy, G.C.: Case studies in just-in-time requirements analysis. In: 2012 IEEE Second International Workshop on Empirical Requirements Engineering (EmpiRE). pp. 25-32. IEEE (2012)
4. Guizzardi, G.: Ontological foundations for structural conceptual models. University of Twente (2005)
5. Guizzardi, G., Falbo, R., Guizzardi, R.S.: Grounding software domain ontologies in the unified foundational ontology (UFO): The case of the ODE software process ontology. In: Proceedings of the XI Iberoamerican Workshop on Requirements Engineering and Software Environments. pp. 244-251 (2008)
6. ISO/IEC 12207:2008, Information technology – Software life cycle processes. pp. 122. International Organization for Standardization, Geneva (2008)
7. JIRA Issue Tracking System, http://www.atlassian.com/software/jira, accessed 08.05.2014
8. Kagdi, H., Collard, M.L., Maletic, J.I.: A survey and taxonomy of approaches for mining software repositories in the context of software evolution. Journal of Software Maintenance and Evolution: Research and Practice 19, 77-131 (2007)
9. Kagdi, H., Maletic, J.I., Sharif, B.: Mining software repositories for traceability links. In: ICPC'07. pp. 145-154. IEEE (2007)
10. Schneider, K.: Experience and Knowledge Management in Software Engineering. Springer, Berlin-Heidelberg (2009)
11. Shekhovtsov, V., Mayr, H.C., Kop, C.: Harmonizing the Quality View of Stakeholders. In: Mistrik, I., Bahsoon, R., Eeles, R., Roshandel, R., Stal, M. (eds.): Relating System Quality And Software Architecture. Elsevier (2014, in print)
12. Shekhovtsov, V.A., Mayr, H.C.: Towards Managing Understandability of Quality-Related Information in Software Development Processes. In: Murgante, B., Misra, S., Carlini, M., Torre, C., Nguyen, H.Q., Taniar, D., Apduhan, B., Gervasi, O. (eds.): Proc. ICCSA'14. Lecture Notes in Computer Science. Springer (2014, in print)
13. Shekhovtsov, V.A., Mayr, H.C., Kop, C.: Towards Conceptualizing Quality-Related Stakeholder Interactions in Software Development. In: Mayr, H.C., Kop, C., Liddle, S., Ginige, A. (eds.): Information Systems: Methods, Models, and Applications, Lecture Notes in Business Information Processing, Vol. 137, pp. 73-86. Springer, Berlin-Heidelberg (2013)
14. Thung, F., Lo, D., Jiang, L.: Automatic defect categorization. In: WCRE'12. pp. 205-214. IEEE (2012)
15. Vandecruys, O., Martens, D., Baesens, B., Mues, C., De Backer, M., Haesen, R.: Mining software repositories for comprehensible software fault prediction models. Journal of Systems and Software 81, 823-839 (2008)

Path-Based Semantic Annotation for Web Service Discovery

Julius Köpke, Dominik Joham, and Johann Eder

Department of Informatics-Systems, Alpen-Adria-Universität Klagenfurt, Austria
{julius.koepke|johann.eder}@aau.at, dominik.joham@gmail.com
http://isys.uni-klu.ac.at

Abstract. Annotation paths are a new method for semantic annotation which overcomes the limited expressiveness of semantic annotations by concept references as defined in the SAWSDL standard. In this work we show some preliminary evaluation of the feasibility of annotation paths for web service discovery. The experiments suggest that annotation paths can capture the semantics of XML schemas and web service descriptions more precisely and appears as a promising approach for improving web service discovery.

Keywords: Web Service Matching, Service Discovery, Semantic Annotation, SAWSDL, Annotation Paths, XML-Schema Matching, Semantic Matching

1 Introduction

Web service discovery aims at (semi-)automating the search for suitable web services. A web service discovery systems accepts a service request (a specification of the needed web service) and a set of web service descriptions (advertisements) as input and returns a list of web service descriptions ranked by relevance for the request. There are many different approaches ranging from the structural or lexical comparison of requests and advertisements to approaches that are based on the explicit definition of the semantics using ontologies [12]. We specifically address the usage of external knowledge provided by semantic annotations with a reference ontology using SAWSDL [6] annotations. The W3C recommendation SWASDL (Semantic Annotations for WSDL and XML-Schema) specifies a light-weight approach for the annotation of web services with arbitrary semantic models (e.g. ontologies). SAWSDL introduces additional attributes for XML-Schema and WSDL-documents. *ModelReferences* refer to ontology concepts and *Lifting-* and *Lowering-Mappings* refer to arbitrary scripts that transform the inputs and output XML-data to and from instances of some semantic model. $ModelReferences$ are proposed for service discovery, while *Lifting-* and *Lowering-Mappings* are proposed for service invocation and only apply to the annotation of inputs and outputs defined by XML-Schema.

Our previous work focused on the annotation of XML-schemas with reference ontologies in order to automate the generation of executable schema mappings

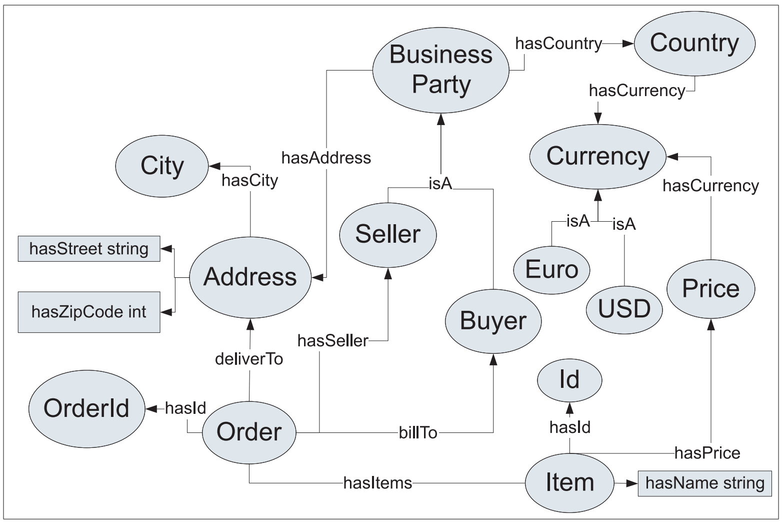


Fig. 1. Example reference ontology [8]

for document transformations [8–10, 7]. We could show that the expressiveness of SAWSDL is not sufficient for the generation of schema mappings when general reference ontologies are directly used for the annotation. Therefore, we have proposed an extended annotation method that is based on annotation paths rather than single concept annotations. Since this method already showed its usefulness for XML-document transformations [7] we assume that annotation paths can also improve web service discovery. The general hypothesis is that if the annotation method allows a more precise definition of the semantics then the precision of service matching for service discovery can be improved. Existing approaches for SAWSDL based service discovery such as [5, 3] can partly solve the problem of non precise semantic annotations by using additional dimensions such as structure or textual similarity.
To give a first answer on this hypothesis we discusses the usage of annotation paths for web service discovery and report some preliminary results.

2 Annotation Path Method

In some examples we show limitations of simple references to concepts for the annotation of arbitrary XML-schemas or web service descriptions with existing reference ontologies. We then present the general concept of annotation paths (for details we refer to [8]).

2.1 Example

The SAWSDL [6] standard addresses semantic annotations for both web service descriptions and for XML-schemas which are related since WSDL description

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/
  <xs:element name="order" sawsdl:modelReference="/order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BuyerZipcode"/>
        <xs:element name="BuyerStreet"/>
        <xs:element name="BuyerCity" sawsdl:modelReference="City"/>
        <xs:element name="BuyerCountry" sawsdl:modelReference="Country"/>
        <xs:element name="SellerCountry" sawsdl:modelReference="Country"/>
        <xs:element name="Item" maxOccurs="unbounded" sawsdl:modelReference="Item">
          <xs:complexType>
            <xs:attribute name="ID" use="required" sawsdl:modelReference="Id"/>
            <xs:attribute name="Name" use="required"/>
            <xs:attribute name="Price" use="required" sawsdl:modelReference="Price"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fig. 2. Sample XML-Schema with model-references [8]

use XML-Schema to define the inputs and outputs of operations. The XML-Schema document shown in Figure 2 is annotated using simple concept references referring to the ontology shown in Figure 1 using the $sawsdl : ModelReference$ attribute.

The annotated document in Figure 1 exhibits the following problems:

- The elements *BuyerZipcode* and *BuyerStreet* cannot be annotated because the *zip-code* is modeled in form of a data-type property and not by a concept in the ontology.
- The *BuyerCountry* element is annotated with the concept *country*. This does not fully express the semantics because we do not know that the element should contain the country of the buying-party. In addition the *Seller-Country* element has exactly the same annotation and can therefore not be distinguished.
- The attribute *Price* is annotated with the concept *Price*. Unfortunately this does not capture the semantics. We do not know the subject of the price (an item) and we do not know the currency.

We have always used exactly one concept for the annotation in the example. However, SAWSDL supports lists of concepts in the $modelReference$ attribute but it does not allow to specify the relations between the concepts in this list. Therefore, this does not help to solve the shown problems. In the examples above we have only annotated data-carrying elements. If we would in addition also annotate the parent elements in this case the *order* element we could add a bit more semantic information. It would be clear that the annotations of the

```xml
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:sawsdl="http://www.w3.org/ns/sawsdl" elementFormDefault="qualified" attril
  <xs:element name="order" sawsdl:modelReference="/Order">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="BuyerZipcode" sawsdl:modelReference="/Order/deliverTo/Address/hasZipCode"/>
        <xs:element name="BuyerStreet" sawsdl:modelReference="/Order/deliverTo/Address/hasStreet"/>
        <xs:element name="BuyerCity" sawsdl:modelReference="/Order/deliverTo/Address/hasCity/City"/>
        <xs:element name="BuyerCountry" sawsdl:modelReference="/Order/billTo/Buyer/hasCountry/Country"/>
        <xs:element name="SellerCountry" sawsdl:modelReference="/Order/hasSeller/Seller/hasCountry/Country"/>
        <xs:element name="Item" maxOccurs="unbounded" sawsdl:modelReference="/order/hasItems/Item">
          <xs:complexType>
            <xs:attribute name="ID" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasId/Id"/>
            <xs:attribute name="Name" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasName"/>
            <xs:attribute name="Price" use="required" sawsdl:modelReference="/Order/hasItems/Item/hasPrice/Price[hasCurrency/Euro]"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Fig. 3. Sample XML-Schema document with Annotation-Path Method [8]

child-elements of the order-element can be seen in the context of an order. Unfortunately this would not help for the ambiguities between the *BuyerCountry*- and the *SellerCountry* element. In general it would require a strong structural relatedness between the ontology and the annotated XML-Schema or service description which we cannot guarantee when many different schemas or services are annotated with a single reference ontology. In addition SAWSDL does not define that there are any relations between the annotations of parent and child elements. A solution for these non precise annotations is the usage of a more specific reference ontology, which contains concepts that fully match the semantics of each annotated element. For example it would need to contain the concept *InvoiceBuyerCountry* and *InvoiceBuyerZipCode*. However, enhancing a general reference ontology with all possible combinations of concepts leads to a combinatorial explosion.

2.2 Annotation Path Method

We propose a new annotation method based on annotation path expressions that are sequences of steps referring to concepts and properties of a reference ontology. The first step of an annotation path is always a concept. The last step of an annotation path can be a concept or a data-type property. Between two concept steps there is always an object property step. Concept steps can have constraints denoted in square brackets. In Fig. 3 we give some examples for annotation paths - more details can be found in [8].

Annotation paths can automatically be represented in form of OWL2 concepts which can be used to extend the reference ontology. For example the path *Order/billTo/Buyer[Mr_Smith]/hasCountry/Country* is represented as a subclass of *Country* that has an inverse *hasCountry* relation to a *Buyer* whose name is *Mr. Smith* who has an inverse *billTo* relation to an *Order*. This can be represented by the OWL expression *Country and inv (hasCountry) some (Buyer and {Mr_Smith} and inv (billTo) some (Order))*.

The extraction of annotations from a schema requires to rewrite the schema in order to cope with reused elements first. The resulting schema may contain additional annotations. Since schema elements can refer to other schema elements and types, the full annotation path has to be concatenated from the annotation paths of the elements. For an example an XML-element *DeliveryAddress* is itself annotated with the annotation path */**Order**/deliverTo/**Address***. It has a type definition *address*. The address type itself contains various elements. One of them is *street* which is annotated with */**Address**/hasStreet*. In order to construct the complete semantics of the *street* element which is a child element of *DeliveryAddress* we get the additional path */**Order**/deliverTo/**Address**/hasStreet*.

3 Path-Based Service Matching Prototype

To apply the annotation path method to web service discovery we implemented a logics based service matcher [1] that operates only on path-based annotations of the inputs and outputs of operations. No other dimensions of the service descriptions are used for matching. The assumption is that two operations with the same inputs and outputs are likely to be the same operation. We do not address the annotation of operations themselves. In order to rank different web services according to a request we automatically generate one XML-Schema for the inputs and one XML-Schema for the outputs of each operation of the advertisements and the request. These schemas are then matched and an overall confidence value for the service match in the interval [0..1] is computed. The ranking is then based on the confidence values. The matching process of the schemas operates in 4 phases:

- *Annotation Path Extraction:* The input and output schemas of each operation are transformed to an internal tree representation where no types are reused using the COMA3[11] library. The annotation paths are rewritten as described in the last paragraph of Sect. 2.2. Finally all annotation paths are extracted from the resulting tree.
- *Extended Ontology Generation:* The annotations are transformed to OWL concepts and an extended reference ontology is created.
- *Matching and Mapping:* The XML-Schemas of the request and of each advertisement are matched based on the annotations using a standard OWL reasoner (pellet). Two schema elements $s1$ from the source schema and $t1$ from the target schema match if the annotation concept (the corresponding annotation path represented as an OWL concept) of $s1$ is equivalent to the annotation concept of $t1$ or if there is a subclass or superclass relation between $s1$ and $t1$. In case of equivalence the confidence value of the match is 1. In case of the subclass match the confidence value of the match is 0.8 weighted by the concept distance between the annotation concept of $s1$ and $t1$ in the extended reference ontology. In case of a superclass to subclass match the confidence value is 0.6 also weighted by the distance in the ontology. After the confidence values are computed for each combination of

elements of the source and target schema, a schema mapping is created based on the best matching elements.
- *Ranking:* Finally, an overall confidence value of each schema mapping is computed by aggregating the confidence values of the mapping elements using min, max or avg. strategies and the advertisements are ordered descending by the overall confidence values.

4 Evaluation

The goal of the evaluation is to provide preliminary results whether the annotation path method leads to better results in service discovery. Therefore, we have evaluated [1] our service matcher that exploits only path based semantic annotations against existing SAWSDL-based service matchers. The assumption is that when this simple service matcher can compete with state of the art service matchers that exploit far more aspects of a service and use advanced techniques such as machine learning, then the usage of annotation paths is also promising for service discovery.
We have annotated a subset of the SAWSDL-TC3[1] data-set with our annotation path method and have evaluated our matcher against service matchers that took part in the *International Semantic Service Selection Contests*[2]. We have evaluated two scenarios:

- *Scenario 1:* The goal of this scenario was to evaluate how, our simple matcher can compete against current state of the art matchers based on existing requests and advertisements of the SAWSDL-TC-3 data-set. In the first scenario we have selected one arbitrary request (*book-price*) and 40 advertisements and have annotated them manually using the annotation path method. Our matcher operated on requests and advertisements which are annotated with annotation-paths and the reference matchers used the original annotations and advertisements of the TC-3 data-set.
- *Scenario 2:* The goal of the second scenario was to asses how our matcher competes against other matchers if the semantics cannot be expressed by simple concept annotations. In this case matchers operating on simple concept annotations can only infer the missing semantics by exploiting other dimensions such as the structure or naming of elements. The second scenario was also evaluated using existing advertisements of the SAWSDL-TC3 data-set. We have only changed the request. We now require for the price of books in *Euro* but excluding tax and we restrict the input to science fiction comics. This cannot be expressed with the used ontologies because no such concepts exists. However, a hint for the standard SASWSDL matchers was provided by the requested output type (*EuroPriceExcludingVAT*) and input type (*ScienceFictionComic*).

[1] http://projects.semwebcentral.org/projects/sawsdl-tc/
[2] http://www-ags.dfki.uni-sb.de/ klusch/s3/index.html

We have executed the evaluation with the Service Matchmaker and Execution Environment (SME2[3]) which is also used for the International Semantic Service Selection Contests. Due to the partial TC3 data-set we were not able to execute all matchers. However, we could execute two major representatives iSem[3, 4] and SAWSDL-MX[5]. The *iSem* matcher when applied for SAWSDL is a hybrid service matcher exploiting inputs and outputs and service names that employs strict and approximated logical matching, text-similarity-based matching and structural matching and automatically adjusts its aggregation and ranking parameters using machine learning. It reached the best binary precision in the contest of 2012. SAWSDL-MX is a typical representative of a hybrid matcher using logics and syntax-based matching. The SAWSDL-TC3 data-set is annotated with relevance grades for each combination of advertisements and requests. A relevance grade is a value between 0 and 3, where 0 stands for not relevant and 3 stands for highly relevant. We used the original relevance grades for Scenario 1 and asked an independent expert to provide the relevance grades for Scenario 2. We have assessed the overall performance of each matcher based on the reached Normalized Discounted Cumulative Gain[2] (NDCG) which is also used in the International Semantic Service Selection Contests. The results of both scenarios are shown in Table 1. Our matcher performed more than 4 percent better than the *SAWSDL-MX* matcher and around 1 percent less precise than the nearly perfect *iSEM* matcher. In the second scenario our matcher performed around 9 percent better than *iSem* and around 12 percent better than *SAWSDL-MX*.

Table 1. Result Comparison

Matcher	NDCG Scenario 1	NDCG Senario 2
Path-Based Matcher	0.977	0.970
iSem Hybrid	0.990	0.886
SAWSDL-MX	0.937	0.867

While these preliminary results do not yet allow to draw final conclusions the annotation paths approach is promising for improving web service discovery. Our simple path-based matcher could clearly show its advantage in Scenario 2 and in Scenario 1 it could compete well with existing state of the art matchers which use far more advanced matching methods and additional aspects of service descriptions and advertisements.

5 Conclusions and Future Work

The annotation path method for semantic annotation has been developed to overcome limitations in the expressiveness of simple concept references. We showed in some feasibility tests that already a simple implementation of an

[3] http://projects.semwebcentral.org/projects/sme2/

annotation path based XML-schema matcher used for comparing web service advertisements with service requests can successfully compete with state-of the art web service discovery systems. We therefore conclude that annotation paths are well suited for capturing the semantics of objects in much finer detail and that the annotation path method and matchers based on it are promising approaches to improve web service discovery. Encouraged by the promising results we plan to evaluate our annotation path based matcher with a larger data-set and against additional existing matchers. Other future work is to integrate our matcher into existing state of the art matchers to gain even better results. Another direction of future work is to evaluate not only the matching precision but also the minimum amount of manual work to semi-automatically create annotation path annotations in comparison to simple concept references.

References

1. Dominik Joham. Path-based semantic annotation of web service descriptions for improved web service discovery. Master-thesis, AAU Klagenfurt, February 2014.
2. Jaana Kekäläinen. Binary and graded relevance in IR evaluations-Comparison of the effects on ranking of IR systems. *INFORM PROCESS MANAG*, 41(5):1019 – 1033, 2005.
3. M. Klusch and P. Kapahnke. isem: Approximated reasoning for adaptive hybrid selection of semantic services. In *Semantic Computing (ICSC), 2010 IEEE Fourth International Conference on*, pages 184–191, Sept 2010.
4. Matthias Klusch and Patrick Kapahnke. The isem matchmaker: A flexible approach for adaptive hybrid semantic service selection. *Web Semantics: Science, Services and Agents on the World Wide Web*, 15(3), 2012.
5. Matthias Klusch, Patrick Kapahnke, and Ingo Zinnikus. Hybrid adaptive web service selection with sawsdl-mx and wsdl-analyzer. In *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 550–564. Springer Berlin Heidelberg, 2009.
6. Jacek Kopecký, Tomas Vitvar, Carine Bournez, and Joel Farrell. Sawsdl: Semantic annotations for wsdl and xml schema. *IEEE Internet Comput.*, 11(6):60–67, 2007.
7. Julius Köpke. *Declarative Semantic Annotations for XML Document Transformations and their Maintenance*. Phd-thesis, AAU Klagenfurt, March 2012.
8. Julius Köpke and Johann Eder. Semantic annotation of xml-schema for document transformations. In *Proc. of OTM'10 Workshops*, volume 6428 of *LNCS*, pages 219–228. Springer, 2010.
9. Julius Köpke and Johann Eder. Semantic invalidation of annotations due to ontology evolution. In *Proc. 2011 of OTM'2011*, volume 7045 of *LNCS*, pages 763–780. Springer, 2011.
10. Julius Köpke and Johann Eder. Logical invalidations of semantic annotations. In *Proc. of CAiSE'12*, volume 7328 of *LNCS*, pages 144–159, 2012.
11. Sabine Maßmann, Salvatore Raunich, David Aumüller, Patrick Arnold, and Erhard Rahm. Evolution of the coma match system. In *OM*, 2011.
12. Debajyoti Mukhopadhyay and Archana Chougule. A survey on web service discovery approaches. In David C. Wyld, Jan Zizka, and Dhinaharan Nagamalai, editors, *Advances in Computer Science, Engineering & Applications*, volume 166 of *Advances in Intelligent and Soft Computing*, pages 1001–1012. Springer Berlin Heidelberg, 2012.

Analyzing Engineering Contributions using a Specialized Concept Map

Arnon Sturm[1,2], Daniel Gross[1], Jian Wang[1,3], Eric Yu[1]
University of Toronto[1], Ben-Gurion University of the Negev[2], Wuhan University[3]

sturm@bgu.ac.il, daniel.gross@utoronto.ca,
jianwang@whu.edu.cn, eric.yu@utoronto.ca

Abstract. Identifying open problems in an engineering domain is a first step towards making new contributions. To identify problems one often examines existing solutions to recognize opportunities for advances. As the knowledge in a domain grows and multiplies, it becomes increasingly difficult to keep track of advances made, especially in relation to evolving needs. Drawing from goal-oriented requirements engineering, we propose a specialized use of concept maps to map out contributions to problem-solving knowledge in an engineering domain. We illustrate the approach using the domain of Architecture Description Languages (ADLs) and discuss usefulness and usability of the specialized concept map.

Keywords: Knowledge Mapping, Requirement Engineering, Concept map

1 Introduction

The state of the art in engineering related fields is a fast moving target. With innovation occurring globally at fast pace, researchers and practitioners must expend significant efforts to keep up with the current state of the art, which is also a prerequisite to pushing the boundaries to better deal with new problems and needs.

To stay up-to-date and to make new contributions, researchers and practitioners must over time maintain an overview of a field, understand the problems addressed and solutions proposed, as well as identify the outstanding issues that should receive further attention. Given the fast pace of new developments, keeping such an overview is challenging. Researchers make use of a number of approaches to consolidate and better understand a research domain. They mainly use literature reviews, including systematic reviews [5], and tagging and classifications approaches[1] [6]. In addition, some research has been done to improve on aforementioned approaches such as to consolidate scholarly works using concept maps [8], cause maps [4], and claim-oriented argumentation [9]. As researchers are looking for innovation, they would like to have supporting tools that would help to cluster related topics, to explicate prob-

[1] There are also reference management systems like EndNote and Mendeley that support classification using folders and tagging.

lems and solutions, to represent and reason about differences between existing solutions, and to analyze how the knowledge in a domain evolves over time. Indeed, the aforementioned approaches offer different kinds of textual, conceptual and visual mapping over domains, however, analyzing their capabilities, it seems that they lack essential capabilities to evaluate or compare state-of-the art studies. Table 1 presents a (subjective) comparison of the approaches with respect to the needed capabilities.

Table 1. Comparing mapping techniques

Approach	**Clustering**	**Expressiveness**	**Reasoning**	**Dynamic Evolution**
Literature Survey	+	+	-	+
Classification	+++	-	-	+
Cause Map	++	++	++	+++
Concept Maps	-	+	-	+++
Argumentation	-	++	+++	+++

Based on our observation that contributions to the state of the art can often be characterized as *means* that come to address specific *ends* in some better way, and the lack of such a view in current approaches, we were motivated to seek an approach based on goal-oriented requirements engineering (GORE) approaches [11] in general and the i* (pronounced i-star) goal-oriented modeling framework approach in particular [11]. The i* approach has at its center the means-ends relationship, and the capability to differentiate alternate means towards some end by indicating their differing contributions towards desired quality objectives. Following that approach and based on our previous work [3], we propose a knowledge mapping technique to represent and map out problems and solutions in engineering domains. We envision that using such a technique would better support researchers in representing and reasoning about research advancements in engineering domains. In this paper, we describe the technique, illustrate its use for the domain of software architecture description languages (ADLs), present initial evaluations and further discuss our vision in developing the technique.

In Section 2 we briefly review the ADL landscape and point to some of the issues we observed. In Section 3 we define the mapping technique and demonstrate it using examples from the ADL domain. In Section 4 we further discuss the proposed technique and reflect on its use. In Section 5 we conclude and elaborate on our vision regarding the usage of such knowledge maps.

2 The Architecture Description Language Domain

The motivation for our research is the observation that much of the knowledge underpinning engineering domains can essentially be characterized using means-ends relationships and qualifying properties. Such a characterization supports systematically representing needs and problems and their linkages to proposed solutions that come to address those problems in some better way. Capturing problems, solutions and such linkages between them helps in systematically identifying what the problems in a domain are, which of those have been addressed to date and how, as well as what the outstanding issues are that could benefit from future exploration.

As an example we looked at the domain of architectural design as one of the important areas in IS design. Specialized languages for supporting architectural design has been an active research area for some time, yet their industry adoption has been limited. In a recent survey that aimed to identify what architects need from architectural description languages (ADLs) [7], the authors identified 150+ ADLs that were proposed over the last decade and half and asked practitioners, amongst others, which ADLs they used, what ADL features they found they need, and more generally what their ADL needs are during architectural development.

One surprising outcome of the survey was the limited adoption of the proposed ADLs in industry. Only a hand-full of ADLs, and mainly those that originated from industry, were in active use, with UML used as an ADL by 86% of the respondents. Furthermore, 86% of respondents indicated that ADLs needed to be extended to meet project-specific needs. Yet, only about 25% of UML users actually extended UML, such as by use of profiles, to meet project-specific needs, with about 73% of respondents using UML as-is, despite its lack of architectural description features.

To better understand the reason of UML adoption, and in particular why it was mostly adopted in organizations as-is without extensions, we mapped out portions of the ADL domain to capture how ADL authors perceived problems and solutions in that domain. Our aim was to identify how a knowledge structure based on means-ends relationships could more systematically tie ADL research to practitioners' needs. While addressing architects needs is clearly an important objective for ADL designers, creators of ADLs nevertheless typically focused on specific technical features they perceived architects would need and offered interesting representational or analytical features, which in the end were however not adopted by practicing architects in industry.

To further examine knowledge structures in the ADL domain we also turned to ADL literature reviews, which consolidate, compare and contrast ADLs. Such literature surveys offer useful perspectives on the knowledge structure of the ADL domain. However, such reviews mainly focused on comparing the feature sets of ADLs using predefined features or feature categories perceived to be of value to the survey authors [2].

We thus aim to complement such textual survey approaches with a conceptual knowledge map that helps characterize and clarify the problems and needs, and different solution approaches in engineering domains, and helps link high level needs and problems to the solution approaches.

3 Specializing Concept Maps for Engineering Domains

During our research we applied the proposed technique to map out portions of a variety of engineering domains including agent-oriented software engineering, geoengineering, web mining, and documenting software architectures. Distilling our modeling experience we adopted a minimal set of modeling constructs including two main types of nodes and several types of links. By convention, the map is laid out with problems or objectives at the top and solutions at the bottom.

- The **task** is the main element in the means-ends hierarchy. A task can be interpreted either as a problem (in relation to lower elements) or a solution (in relation to higher elements). It is named with a concise verb phrase typically, and graphically depicted as a rectangular shape with rounded corners. For example, in Figure 1, which illustrates part of the ADL knowledge map, the task "Define Architecture" is a problem to be addressed. It can be achieved by the tasks of "Define New Architecture" or "Utilize Existing Knowledge". Both solutions can in turn can also be viewed as sub-problems that need further addressing. A task can have associated contexts and a set of references (i.e., the knowledge sources) justifying the existence of such a task within the map. These are not shown in Figure 1 to avoid clutter, but are supported by the tool.
- A **quality** element is used to express quality attributes that are desired for associated tasks. Examples in Figure 1 include "Scalable", "Traceable", and "Architecture Quality". A quality is depicted as an ellipse, and is typically named with adverbial or adjectival phrases or quality nouns (e.g., "-ilities").
- Links connect tasks and qualities. In the following we elaborate on the link types.
  - The **achieved by** link represents a means-end relationship. The arrow points from the "end" to the "means". Figure 1 indicates that "Use ADL" is one way of achieving "Design New Architecture". "Use WRIGHT" and "Use UML" are alternative ways of achieving "Use ADL".
  - The **consists of** link indicates that a task has several sub-parts, all of which should be accomplished for the parent task to be accomplished. In Figure 1, "Devise Architecture" consists of "Define Architecture", "Select Technology", and "Communicate the Architecture", among other problems that need to be addressed.
  - The **association** link (an unlabeled and non-directional link) indicates the desirable qualities for a given task. These qualities are to be taken into account when evaluating alternative ways for accomplishing the task. For example, "Adoptable", "Extensible", are qualities that could differentiate among different ways of "Use(-ing) ADL".
  - The **extended by** link indicates that the target task is an extension of the source task. For example, "Create UML-Profile" is an extension of "Use UML". All qualities that hold for the parent task also hold for its extensions.
  - The **contribution** link (a curved arrow) indicates a contribution towards a quality, from a task or another quality. The contribution can be from strong negative to strong positive contribution, which are determined subjectively by the map creator based on the available resources. For example, "Use UML" is well contributed ("++") to "Weavable into SDLC", and also contribute ("+") to "Analyzable". The alternative "Use WRIGHT" is well contributed ("++") to "Formal" and contribute ("+") to "Weavable into SDLC", and "Usable".

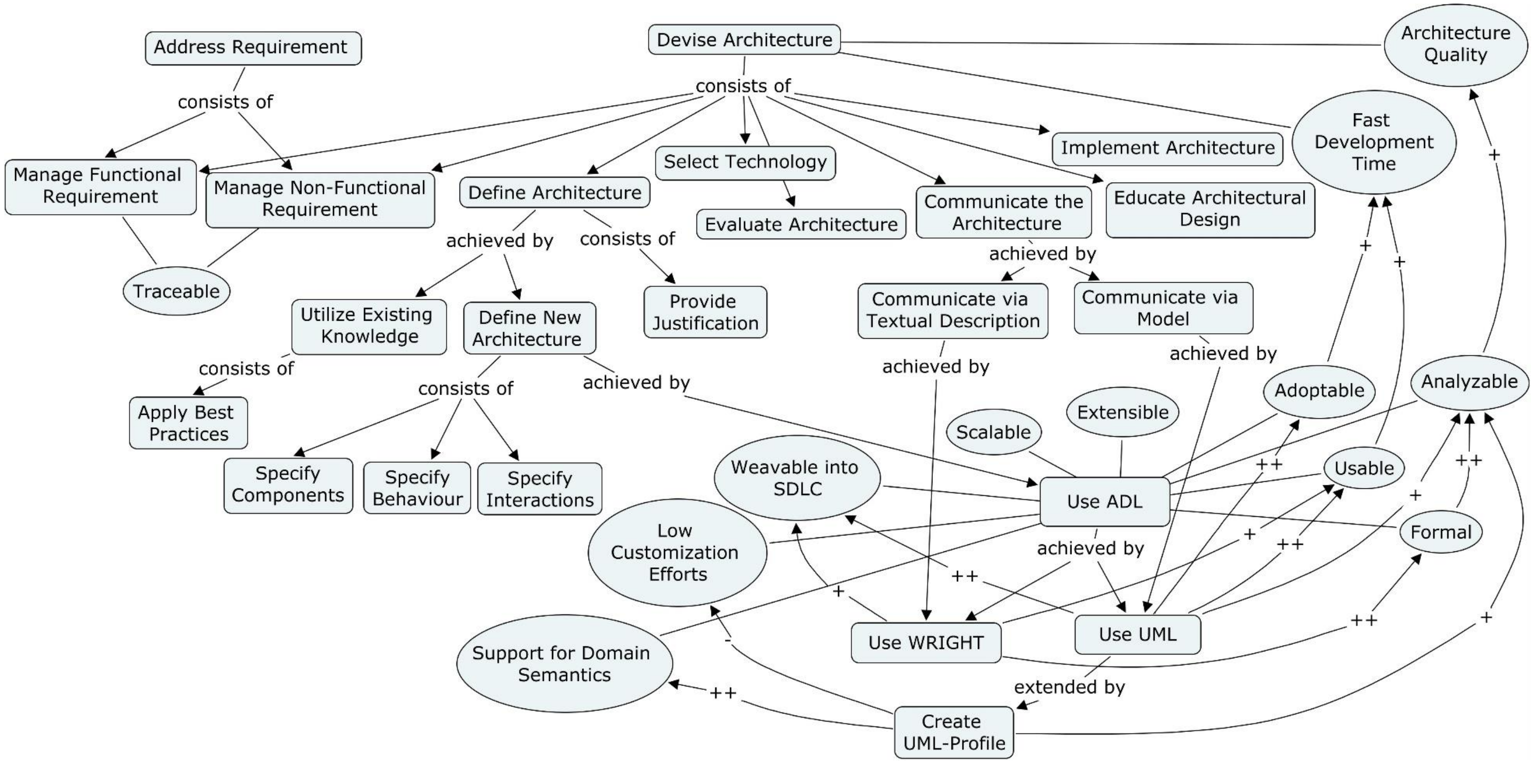


Figure 1. A partial knowledge map of the ADL domain

We used the cmap tool[2] to draw the knowledge map in Figure 1. Using the cmap tool allows us to benefit from all implemented features of the platform, including collaborative modeling and sharing of concept maps.

It should be noted that Figure 1 is a map aims to serve as an index to the actual knowledge. The purpose of the map is not to create new knowledge rather to organize the knowledge in a way that would increase its accessibility.

To construct the map in Figure 1, we analyzed an informal survey of architects duties and skills offered on the CMU SEI website [9] and the aforementioned ADL architects needs survey paper [7]. We further looked at comparison of various ADLs reported in the literature [2]. Following these resources, we were able to construct the map while having supporting evidence for the claims implied by nodes and relationships included in map.

4 Discussion

Using the knowledge map in Figure 1 we can point to the main justifications for choosing one ADL over another by following their level of contribution to the qualities associated with the use of an ADL. Which ADLs were chosen in practice, and why? According to the Malavolta survey [7], most architects adopting UML as an ADL decided to live with the limitations of UML rather than choosing an ADL that fits with their specific software architectural analysis needs, because the generic version of UML was a notation already known in the organization, and hence required minimal learning effort during adoption. This explanation is indicated in Figure 1 by the contribution link from "Use UML" to "Analyzable": using UML has limited contribution to having an analyzable architectural description; a "++" link from "Use UML" to Adoptability: using UML has a strong contribution to being adoptable in the organization; a "++" link from "Use UML" to "Weaveable into SDLC": indicating another advantages over WRIGHT; and the "-" contribution link from "Create UML Profile" to "Low Customization Efforts", indicating that creating a UML Profile contributes negatively to reducing customization efforts, and thus rarely used.

Hence, the concept map helps capture and visualize, using selected concepts and relationships, the arguments that drove many architect's adoption decision-making. More specifically, adoptability in the organization is an overriding concern, which needs to be addressed well before other useful features, are introduced in an ADL.

Another insight for future research that may be derived from the knowledge map is that ADLs only cover a small part of the overall responsibilities of software architects. According to Figure 1 decisions about ADL use mainly support the definition of new architectures, which is, however, only one task of a variety of others that architects are engaged in during architectural development, some of which might be more important to the organization than more formal descriptions of architectures. The knowledge map should thus help researchers and practitioners in seeing the broader picture of needs into which more particular research directions are positioned.

[2] http://cmap.ihmc.us/

While the proposed technique facilitates representing problems and solutions in existing state-of-the-art, we encountered a number of challenges while self-examining it in the various domains aforementioned and with various viewpoints (e.g., mapping a domain, mapping a specific research, adopting a top-down approach, and adopting a middle-out approach):

Conceptual Mismatch: Identifying problems, solutions, qualities, and the relationships among them is often non-trivial. Researchers and stakeholders often present needs and benefits in solution-oriented terminology and languages and neglect the connection with the problem-oriented aspect.

Naming Decompositions: During the construction of a knowledge map elements are decomposed into lower level elements. Decomposition is the main mechanism to unearth variation and differences in approach details (solution features) that matter with respect to qualities. However, in some domains it appears difficult to identify and name those solution feature “components” that differentiate among alternative approaches. This suggests that more holistic representations of solution approaches, or, that more finer-grained concept map based analysis guidelines to help make explicit in what way proposed solutions differ in their details, may be needed.

Multiple vantage points and terminology use: Because of different viewpoints map creators might take, they may develop maps differently, both in terminology and in the abstraction level. Furthermore, it is in the purview of the map creator to decide which level of abstraction is the most fitting to express problems and solution approaches. When constructing larger maps out of contributions from different map authors, aligning the levels of abstraction is thus non-trivial.

Scalable tool support: Having good tool support is often a key weakness in proposed approaches. Using concept maps we can take advantage of existing tools, and use “scalability” features such as: element expanding/collapsing and map referencing.

Domain knowledge extraction: Currently, knowledge extraction and its mapping are done manually and obviously, subjectively, as implied before. This introduce a burden on adopting the approach. Nevertheless, we envision crowd-mapping as an approach that distributes the burden across interested many participants, who benefit from mutual contributions, and approaches to automated concept extraction from bodies of engineering text guided by the proposed concepts that link needs with solutions.

5 Conclusion and Future work

With the fast moving technological/engineering innovations landscape, new approach proposals that address novel challenges are continuously devised. In this paper we propose a technique to map out research fields using a light-weight modeling technique, based on a well-known concept mapping approach and argue for its benefits, such as the ability to represent and facilitate the analysis for novel proposals, gaps of un-addressed problems, as well as, other kinds of analysis such as tracing of possible reasons for adoption or non-adoption of proposed approaches. We believe that the approach is applicable to any engineering domains that fit into the problem-solution means-end approach. Yet, its benefits depend on the domain maturity.

From preliminary user studies [1], we found that the proposed technique supports examining a domain of research not only from the solution point of view, but also allows for emphasizing problems addressed, and in particular the properties in the problem space that offer significant advantages over prior art. Having both problems and solutions captured in one place helps better reviewing and understanding a domain. We were also able to identify gaps in which further research could be performed.

To further explore and facilitate the use of knowledge mapping, in the future we plan to expand knowledge map capabilities in a number of directions: further develop guidelines for map creators to support extracting knowledge from research domains and including them in knowledge maps; support a crowd-mapping approach whereby different stakeholders could contribute to creating, arguing about and improving on a collaborative knowledge map; support for trust mechanisms, as well as, empirical evidence based additions to knowledge maps that offer additional insights; develop semi-automated reasoning support to identify gaps or even possible solution approaches to identified gaps, across different knowledge maps; and develop automated extraction of knowledge mappings from bodies of engineering texts, guided by core concepts proposed in this paper. Additional evaluations for testing the benefits of the proposed technique are also required.

References

1. Abrishamkar, S.: Goal-Oriented Know-How Mapping, Modelling process documentation, a prototype, and empirical studies, M.Sc. Thesis, University of Toronto (2013)
2. Clements, P.C.: A survey of architecture description languages. Proceedings of the 8th International Workshop on Software Specification and Design, pp.16-25 (1996)
3. Gross, D., Sturm, A., Yu, E.: Towards Know-how Mapping Using Goal Modeling. iStar, 115-120 (2013)
4. Eden, C., Ackermann, F., Cropper, S.: The analysis of cause maps. Journal of Management Studies, 29(3), 309–324, (1992)
5. Kitchenham, B., Brereton, P., Budgen, D., Turner,M., Bailey,J., Linkman, S.: Systematic literature reviews in software engineering - A systematic literature review. Inf. Softw. Technol. 51 (1), 7-15 (2009)
6. Kwasnik, B.: The role of classification in knowledge representation and discovery. Library Trends, 48, 22-47 (1999)
7. Malavolta, I.,Lago, P., Muccini, H.,Pelliccione, P., Tang, A.: What Industry Needs from Architectural Languages: A Survey, IEEE Transactions on Software Engineering, 39 (6), 869-891 (2013)
8. Novak, J. D., Cañas, Al. J.: The Theory Underlying Concept Maps and How To Construct and Use Them, Institute for Human and Machine Cognition (2006)
9. SEI, http://www.sei.cmu.edu/architecture/research/previousresearch/duties.cfm, Shum, S.B., Motta, E., Domingue, J.: ScholOnto: An ontology-based digital library server for research documents and discourse. International Journal of Digital Libraries, 3(3): 237-248 (2000)
10. Yu, E., Giorgini, P., Maiden, N., Mylopoulos, J.: Social Modeling for Requirements Engineering, MIT Press (2011)

Work Systems based Fractal Architecture of Information Systems

Marite Kirikova

Riga Technical University, Latvia
marite.kirikova@cs.rtu.lv

Abstract. Contemporary information systems have to satisfy needs of agile and viable enterprises. They shall include mechanisms of business intelligence, business process management, information technology infrastructure management, and alignment between business and computer systems. The mechanisms for business process handling and computer systems handling are similar, and the mechanisms for their continuous integrated improvement also are similar, therefore the architecture of information systems components that support these processes can have a measure of similarity if considered at a particular level of abstraction. The paper, focusing on aforementioned similarities, uses St. Alter's work systems paradigm for constructing fractal architecture of information systems that can be used for supporting agile and viable organizations.

Keywords: work system, fractal architecture, information systems.

1 Introduction

Cloud solutions and use of business intelligence tools have transformed the landscape of information systems from relatively rigid internal architectures to more flexible and open structures of information handling [1], [2]. This refers to more flexible distribution of physical devices as well as possibility to acquire real time data that can be used for introducing changes in business and information technology solutions. The question arises how these abilities of information technology solutions can be represented in information systems architectures so that complexity that increases with the introduction of higher variability and flexibility could be embraced and managed.

While business, software, and hardware systems are very different, the mechanisms used in their analysis and management do not differ so much. For instance, similarities can be found in the actor based approaches in business analysis and actor based approaches in parallel programming. Moreover, according to our experience [3], [4] - data acquisition and analysis methods for big data analysis in social networks can be compared with similar methods in computer networks. These similarities suggest to seek for common architecture patterns that could be used in business and information technology domains, since the information systems processes cross both domains.

In this paper we propose to use work systems paradigm introduced by St. Alter [5], as a basis for reflecting architecture of information systems. Taking into consideration that information systems concerns both business and information technology subsystems of an enterprise, we use inclusion relationship between different work systems of the enterprise

The paper is structured as follows. We briefly discuss related work in Section 2. In Section 3 we propose the work systems paradigm based model of fractal information systems architecture. In section 4 we discuss the benefits of viewing information systems architecture as a fractal architecture composed of multiple work systems. Brief conclusions and directions of further research are presented in Section 5.

2 Related Work

The approach proposed in this paper is based on two main sources of related work, namely, on the work systems theory [5], and contemporary applications of viable systems model [6], [7], [8].

According to Steven Alter [5] "an information system is a work system whose processes and activities are devoted to processing information, that is, capturing, transmitting, storing, retrieving, manipulating, and displaying information. A work system is a system in which human participants and/or machines perform work (processes and activities) using information, technology and other resources to produce specific products and/or services for specific internal or external customers". The customers can be external customers of the enterprise as well as internal customers (one sub-work system has produced information valuable for another sub-work system). The work system is embedded in its environment, and depends on organizational strategy and infrastructure (see Fig. 1).

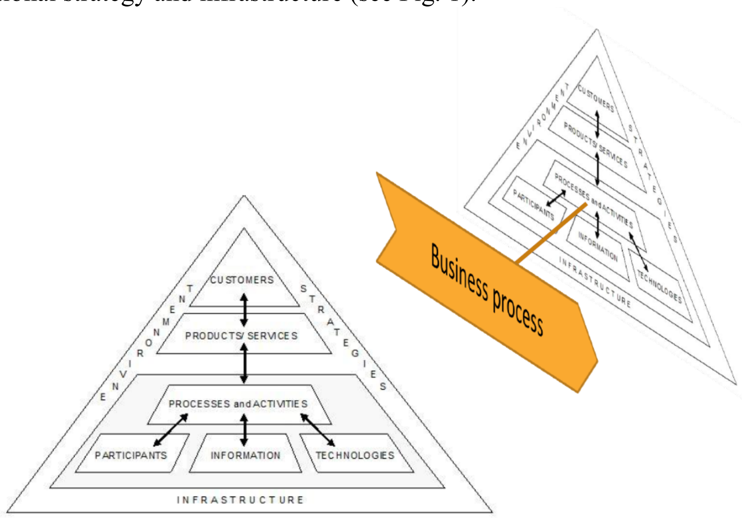


Fig. 1. Work system behind the process (the triangle in the figure is adopted from [5])

Thus, we can state that behind each organizational process there is a work system. From the modeling perspective, there is a real or virtual work system behind of each business process at any level of abstraction or decomposition. On another hand, for each business process there can be find an information processing sub-process (or activities that themselves perform a transformation of inputs into outputs and therefore can be regarded as processes, which produce information). Thus, using Alter's work system paradigm it is possible to identify information systems architecture that consists of work systems that are structured according to the chosen model of representation of business processes. This issue will be discussed in more detail in the next section.

Contemporary applications of viable systems model show that an enterprise has to handle its internal work systems as well as it has to have good environment scanning capabilities at the operational and strategic levels [6], [7], [8]. For the enterprise to be viable, its internal units have to have a measure of autonomy and it should be organized as fractals [9] (in Fig. 2 each "ONE", which corresponds to the processes that bring value to the customer, consists of a smaller scale viable systems model). Such structure enables flexibility that is essential to ensure agility of enterprises. Thus, according to the viable systems model, for the value adding and strategic processes of the enterprise (processes that are directly related to the external environment) we can distinguish at least three sub-processes: production (transforming given inputs into given outputs), environment scanning (operational and/or strategic), and internal control or management (see Fig. 2). In this paper we extrapolate that such three sub-processes are applicable to any process in an enterprise, because also enabling processes and other management processes have to scan their environment inside the enterprise.

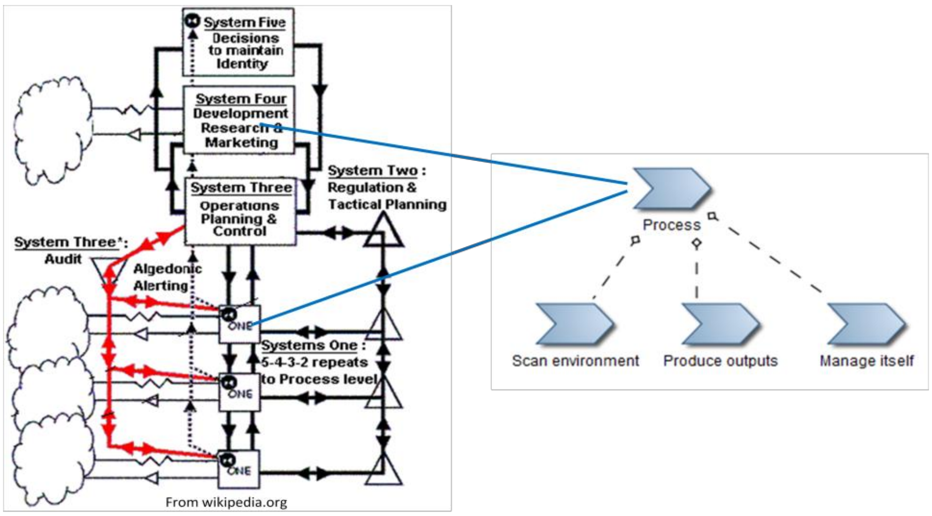


Fig. 2. Viable systems model: a process perspective

Application of viable systems model for contemporary enterprises is a multidisciplinary research topic, which is out of the scope of this paper. Hereby we just borrow the idea of necessity to be aware of external environment, to produce the value, and to be able to manage itself (quality management, change management, etc) for each autonomous unit (we regard the process as an autonomous unit here). Another issue is that while the viable systems model has a fractal architecture, the fractal architecture of information systems proposed in this paper is viewed from the point of view of an information system as a subsystem of an organization, not just from the point of view of operational fractals of the viable systems model.

3 Similarities in Information Systems Support for Organizational Processes.

As described in previous section, the viable systems model indirectly prescribes that each value producing unit has to be aware of its environment, and handle the value production and self-management sub-processes. Each of this sub-process certainly needs information that, in turn, is supported by the information systems sub-processes (see View A in Fig. 3). These information systems sub-processes have to be supported by particular human or artificial information handling units that represent a specific part of information systems architecture. Thus there will be and information systems process behind of each business process as it sub-process (View B in Fig. 3). This information systems process includes information system sub-processes for environment scanning, value production, and self-management.

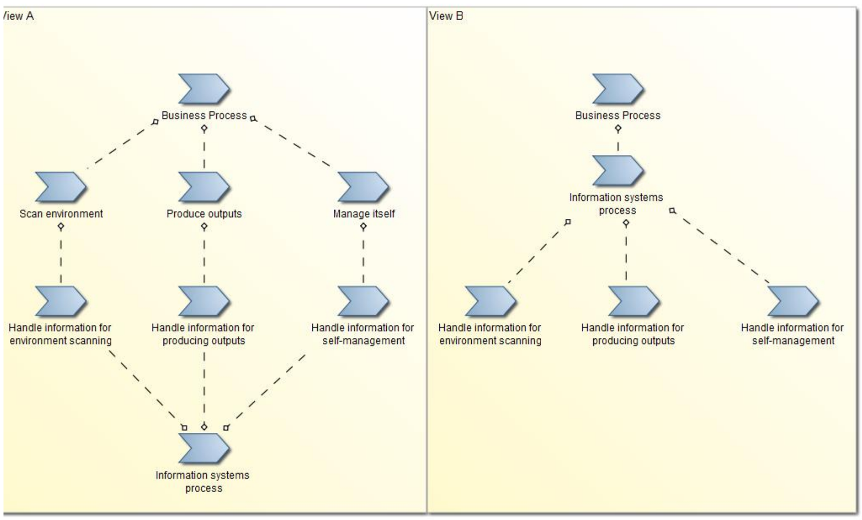


Fig. 3. Information systems process as a sub-process of business process (View A - sub-process level dependency, View B - process level dependency)

The information systems processes are part of work systems that are handling them. Thus behind each information systems process, that supports business process, there is a work system capable of information handling for the particular process (see Fig. 4). The work system in Fig. 4 is a part of work system reflected in Fig 1.

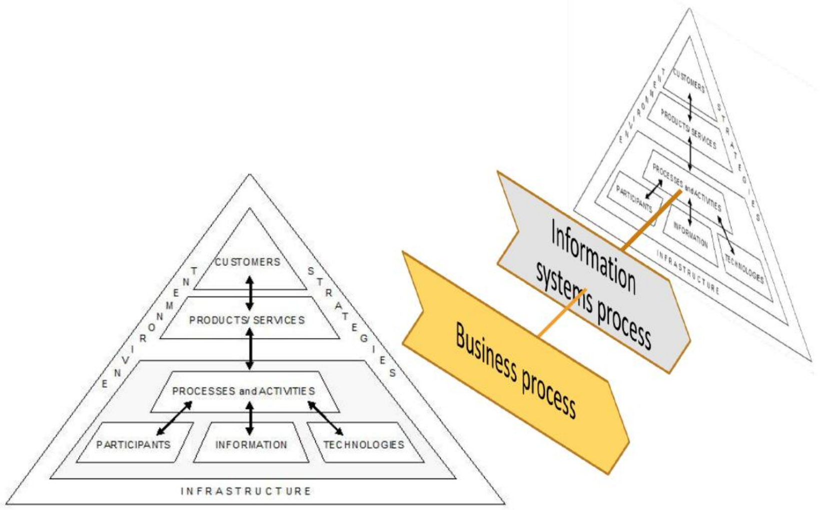


Fig. 4. Information systems architecture work system behind the business process (via corresponding information systems process)

From the process point of view all work systems behind information system processes are self-similar, they have to have a measure autonomy, and they are a part of a whole work system that supports all information processes in a an enterprise. Thus work systems that support business processes form a fractal architecture [7], [9] that can expose a high level of flexibility needed for agile and viable companies.

On another hand, information systems management also is a business process, thus the scanning of information systems environment, information production, and management of information system are sub-processes to be supported (recursively) by the information system. This applies to information systems processes at a high level of abstraction as well to physical processes, such as cloud management or software development process management, or hardware cluster management [1], [2]. In all cases the same type of work system processes are present, just at different level of scale and using information systems architecture elements of different substance (see Fig. 2, Fig. 3, and Fig. 4).

The proposed fractal elements of the information systems architecture can be recognized at two dimensions: from the value production dimension and the work systems substance dimension. At the value production dimension we consider all business process including their variants and value production oriented decompositions. Value production oriented decomposition is different from business process sub-process types reflected in Fig. 2 and Fig. 3. Value production decomposition means product oriented decompositions, e.g., the process "educate students" can be decomposed into sub-processes "educate students in chemistry" and "educate students in physics" (alternatively these can be viewed as process variants, too), the process "manufacture cars" can be decomposed into sub-processes "manufacture sport cars" and "manufacture city cars", or "manufacture engine" and "manufacture navigation system". We use the value production dimension at the business level. At information systems level we use work systems substance dimension, where the work system can be considered as a virtual actor composed from human participants and technologies: software, and hardware elements (Fig. 5) (including different its variants and decompositions still reflecting human actors as part of the work system); software (application software and/or systems software) together with hardware systems; as well as pure hardware systems with embedded information handling functions.

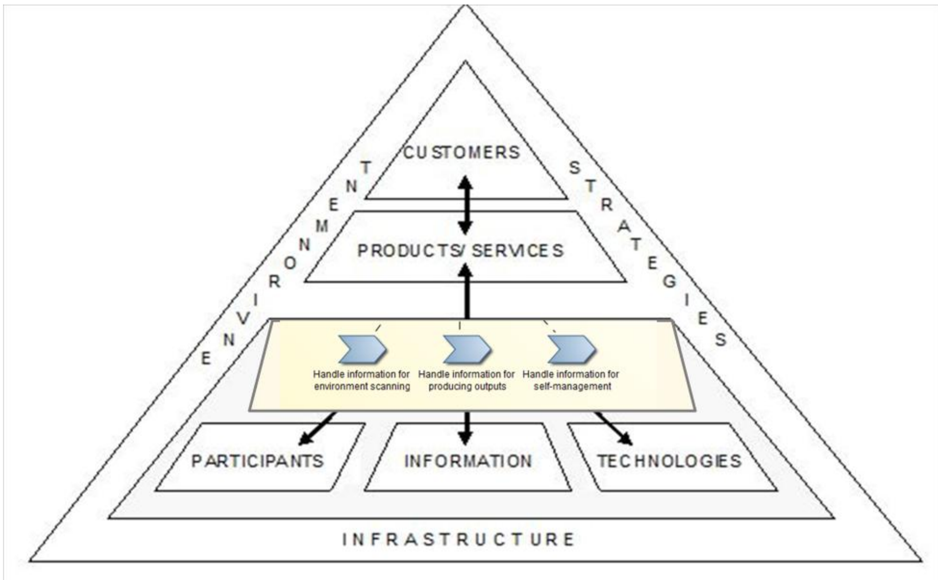


Fig. 5. An element of fractal information systems architecture

The work systems element of information systems architecture, reflected in Fig. 5 suits for both: business value production dimension and work systems substance dimension, from the point of view of information handling.

4 Discussion

There are the following benefits from viewing information systems architecture as a fractal architecture composed of multiple work systems:

- This helps to ensure that all business processes are supported by information system services (sub-processes for environment scanning, value production, and self-management). These services or sub-processes are essential for agile and viable companies as they have to cope with unpredictable changes in their environments.

- Consideration of information systems architecture as a fractal system of work systems gives an opportunity to at least recognize (and if possible - manage) all actual virtual information system work systems supporting the business system regardless of ownership of software and hardware and regardless of functional boundaries of an enterprise.
- Since the work system can be considered as a virtual system, the changes in the architecture can be introduced on a regular basis by combining the work systems or introducing their new versions at the needed level of abstraction or granularity, based on self-similarity principles of fractal systems [7].
- While the particular concept of work system is introduced on the basis of viable systems model (Fig. 2) by considering its operational and strategic processes that have to directly analyze the external environment of an enterprise, still the concept is applicable also for business processes that do not belong to two abovementioned categories. Such approach enables to look beyond not only functional boundaries of an enterprise, but also to consider customers of the enterprise as partners and switch between physical and virtual boundaries of enterprises in modeling of information systems.

Currently the proposed approach of considering information systems architecture as a fractal system of work systems is applied to two library processes in an university. Library processes do not belong to operational or strategic processes directly. Nevertheless they may have a direct relationship with the external environment. We consider two processes - book ordering and acquisition of electronic resources. The information systems work systems for both processes cross several functional units of the university and for both processes environment scanning, value production, and self-management sub-processes are relevant. Similar information technology support is used for external environment scanning and internal environment scanning (e.g. acquisition of usage statistics and acquisition of popularity statistics). The results obtained so far show that a specific organizational processes and work systems based information systems architecture management system is needed to fully benefit from the application of work systems based fractal information systems architecture.

5 Conclusions

The paper proposes to use the concept of a work system for designating virtual and physical components of fractal information systems architecture. The proposal roots in related work on work systems by St. Alter [5] and contemporary applications of viable systems model [6], [7], [8]. However, here the notion of fractality does not exactly consider (but can incorporate) recursive fractality prescribed by viable system model. In the proposed approach, the fractality is considered from the point of view of the value production dimension and the work systems substance dimension. The value production dimension concerns value for external customers as well as value for internal customers. The approach allows to consider customers as partners and to switch between physical and virtual boundaries of enterprises in modeling of information systems. The work systems dimension permits to view the basic information system sub-processes at the level of virtual actors composed of human actors, software, and hardware; at the level of software and hardware; and at the level of hardware only.

The work system is suggested to be considered as an architectural component that for each given business process supports environment scanning, value production, and self-management.

The paper is limited to the introduction of the concept at a high level of abstraction. Two experiments are currently in progress to clarify the details and prepare the ground for defining requirements for work systems based fractal information systems architecture management system. A number of case studies in varieties of contexts will be needed to provide detailed guidelines for application of the proposed concept.

Acknowledgment

This work is partly supported by the research grant of Latvian Council of Science, Project No. Z12.0342.

References

1. Cloud Transforms IT, EMC^2 (2013)
2. Big Data Transforms Business, EMC^2 (2013)
3. Rudzajs, P.: Towards Automated Education Demand-Offer Information Monitoring: the System's Architecture. Selected Papers from Workshops and Doctoral Consortium of the International Conference on Perspectives in Business Informatics Research, BIR2011. pp. 252–265. Springer, Heidelberg (2012)
4. Asnina E.: Essentiality of changes in business models, Proceedings of the 2nd International Business and System Conference BSC 2013, M. Kirikova, G. Lazdane, J. Grabis, K. Lace (Eds.), RTU Press, pp. 44-51, available at https://bsc2013-journals.rtu.lv/issue/current/showToc (2013)
5. Alter, St.: Defining information systems as work systems: implications for the IS field, European Journal of Information Systems, Vol. 17, pp. 448-469 (2008)
6. Hoverstadt, P.: The Fractal Organization: Creating Sustainable Organizations with the Viable Systems Model, Wiley (2008)
7. Espejo, R., Reyes, A.: Organizational Systems. Managing Complexity with the Viable System Model, Springer, Berlin (2011)
8. Rios, J. P.: Design and Diagnosis for Sustainable Organization, Springer, Berlin (2012)
9. Kirikova M.: Towards flexible information architecture for fractal information systems. In: The Proceedings of the International Conference on Information, Process, and Knowledge Management, eKNOW 2009, A. Kusiac and S. Lee (Eds.), IEEE Computer Society, pp 135-140 (2009)

IDECSE: A Semantic Integrated Development Environment for Composite Services Engineering

Ahmed Abid[1] , Nizar Messai[1], Mohsen Rouached[2], Thomas Devogele[1] and Mohamed Abid[3]

[1] LI, University Francois Rabelais Tours, France
ahmed.abid@etu.univ-tours.fr,
{nizar.messai,thomas.devogele}@univ-tours.fr
[2] CCIT, Taif University, Saudi Arabia
m.rouached@tu.edu.sa
[3] CES Laboratory, Sfax University, Tunisia
mohamed.abid@enis.rnu.tn

Abstract. In this paper, we introduce IDECSE a new integrated approach for composite services engineering which is based on Semantic Web and Data Mining. IDECSE considers semantics in all the composition steps: user query, semantic classification of services in the registry, composing services, and verifying the composition process. By considering semantics for describing, discovering, composing, and monitoring services, IDECSE addresses the challenge of fully automating the discovery, composition and monitoring processes while reducing development time and cost. IDECSE appeals for data mining techniques, namely Formal Concept Analysis, for classifying and mining services into service registry in order to anticipate relevant services search and reduce services search space.

Keywords: Web Service Composition, Semantic Web, Data mining.

1 Introduction

Web services are software applications that can be advertised, located and invoked across the Web. Nowadays, an increasing amount of organizations implement their business core and outsource their services over Internet. Thus the ability to effectively select and integrate different services at run-time is an important step towards the development of Web service applications. If no single Web service can satisfy the functionality required by the user, there should be a possibility to create and compose new Web services from existing ones. Considerable academic research and industrial efforts have focused on various aspects of Web service composition ranging from service discovery, to composite service specification and deployment. In this context, important initiatives have been conducted to provide tools and languages that allow an efficient integration of heterogeneous services. Standard languages such as UDDI[4], WSDL[5], and

[4] http://uddi.org/pubs/uddi
[5] http://www.w3.org/TR/wsdl

SOAP[6] were proposed to define standard ways for service discovery, description, and invocation. WSBPEL[7] has focused on representing service compositions where the process flow and bindings between services are known a priori. Later on, following the emergence of the Semantic Web and the fast growth of its related technologies, enhancing Web services description by a semantic level has became one of the basic requirements for efficient services discovery and composition. Since then, several standardization efforts have been done to provide languages which allow to semantically describe Web services on the one hand, and which support efficient automation of the discovery and composition tasks by formal reasoning on services description on the other hand. Standard languages, mainly Web Ontology Language for Services (OWL-S)[8] and Web Service Modeling Ontology (WSMO)[9], were proposed to allow considering semantic aspects in the description and reasoning about services. Based on such languages, many frameworks was proposed for services composition and deployment [1–4]. Despite these efforts and progresses, Web services composition remains a challenging task for the following reasons. First, the number of available services is dramatically increasing. This requires composition frameworks to be accurate, scalable, and reliable to look up and select the most appropriate services for users requirements. Second, services are developed by different organizations based on different types of models and platforms. This heterogeneity in services modeling creates semantic gaps between the presentation of their specification. This requires composition frameworks to provide efficient tools to support bridging such gaps. Third, Web services can be created and updated rapidly. This requires the composition frameworks should be able to dynamically detect and interact with such changes at run-time. Fourth, specifying composition requirements needs the use of high-level languages that are easy to understand, in order to allow end users to express their functional and non-functional requirements in an effective way. Fifth, in case of failure to fulfill user's goal, the composition process should be able to iteratively refining the goal specification in an intuitive way to build composite services. Finally, run-time monitoring and adaptation strategies are primordial to ensure the correctness and the scalability of the composition environments. While numerous composition approaches have been developed, very little has been done towards dealing with these challenges. In this context, this paper introduces IDECSE, Integrated Development Environment for Composite Services Engineering, which considers semantics in all the composition steps: analyzing user query, semantic classification of services in the registry, composing services, and verifying the composition process. This approach aims to provide an easy way to specify functional and non-functional requirements of composite services in a precise and declarative manner, to guide the user through the composition process, while allowing modification or feedback, and finally to enable generating outputs in a deployable language. The rest of the paper is organized

[6] http://www.w3.org/TR/SOAP
[7] http://www-106.ibm.com/developerworks/webservices/library/ws-bpel
[8] http://www.w3.org/Submission/OWL-S
[9] http://www.w3.org/Submission/WSMO

as follows. Section 2, presents the architecture of the proposed framework and details its modules. Section 3 briefly reviews the best known existing approaches before comparing them to IDECSE. Section 4 concludes the paper and outlines current and future work.

2 IDECSE Framework

The IDECSE follows the generic architecture of Web services composition frameworks [5] which contain the following components: Translator, Process Generator, Evaluator, Execution Engine and Service Repository. Figure 1 shows the main parts of a composition framework in a high-level of abstraction (i.e. without considering particular algorithms, languages or platforms). The Service Requester consumes services, following their requirements, offered by service providers, whereas the service provider produces services and puts them into the Service Repository (steps 1 and 2). The Translator translates between the external languages used by the Requester and the internal languages used by the Process Generator (steps 3 and 4). The Process Generator produces plans that combine the available services from the service repository to satisfy the Request (step 5). The evaluator then evaluates all produced plans and returns the best one for execution(steps 6 and 7). Finally, the Execution Engine executes the plan and returns the result to the service requester (step 8).

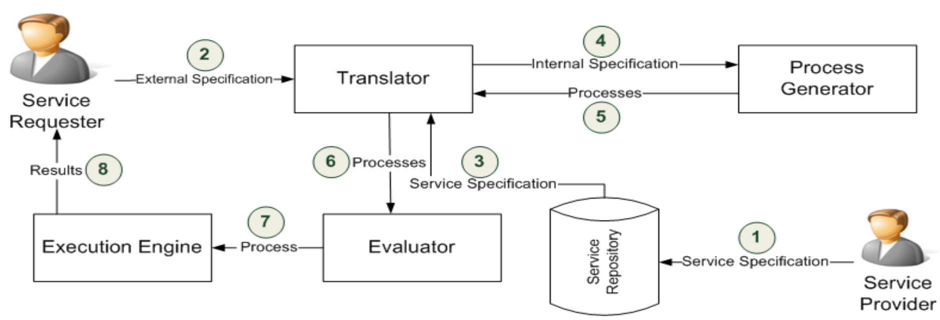


Fig. 1. General framework for Web service composition

IDECSE enhances this architecture to address the challenge of fully integrating semantics in all stages of the composition global life-cycle. First user requirements are more understood using refinement techniques such as generalization or specification of concepts from a given ontology. Second, IDECSE appeals for data mining techniques for classifying and mining services into service registry based on semantic relations. Third, IDECSE is based on two types of reasoners: a similarity-based reasoner and a logic-based one. The IDECSE architecture is depicted in Figure 2. It consists of five modules covering the global composition life-cycle (i.e. specification, classification, composition, deployment, and monitoring). These modules are described in the following sub-sections.

2.1 Service Request Module

The Service Request module translates user requirements to an internal language to be used either by the Service Classification module or the Service Reasoning module. The Graphical Query Editor relies on domain ontology to "understand" the requirements before enriching them through adding new ontology concepts based on semantic relations such as generalization, specialization, etc. The Query is then parsed to extract functional and non-functional requirements. Functional requirements are modeled using the IOPE Extractor, which extract the Input, Output, Precondition and Effects. Non-functional requirements are specified as QoS parameters. Extracted user requirements are then modeled as a new requested service called S_R. Given a domain ontology O, a user query Q modeled as S_R, consists of a set of provided inputs $S_{R_{in}} \subseteq O$, a set of desired outputs $S_{R_{out}} \subseteq O$, a set of preconditions $S_{R_{pre}} \subseteq O$, a set of effects $S_{R_{eff}} \subseteq O$, and a set of quality of service constraints $S_{R_{qos}} = \{(q_1, v_1, w_1), (q_2, v_2, w_2), ..., (q_k, v_k, w_k)\}$, where $q_{i(i=1,2,...,k)}$ is a quality criterion, v_i is the required value for criterion q_i, and w_i is the weight assigned to this criterion such that $\sum_{i=1}^{k} w_i = 1$, and k the number of quality criteria involved in the query. We can model S_R as $S_R = \sum IOPE + \sum QoS$.

2.2 Service Classification Module

To deal with the important number of Web services and instead of considering the whole service registry, this module allows to classify available services semantically into classes according to their similarities. Its second role is to return only relevant services to S_R from the registry. The module contains four components which are Service Projector, Service Description Extractor, Service Similarity Calculator and Relevant Service Selector. The Service Projector selects services capabilities based on syntactical and semantic description for each service (i.e. WSDL and OWL-S description for each service) into one interface. The Service Description Extractor, extracts useful parameters from service capabilities. The Service Similarity Calculator Sub-Module is based on data mining techniques for classifying services into classes according to their relevance and similarity in order to anticipate relevant services search and reduce services search space. The Formal Concept Analysis (FCA) [6] formalism and its extension to complex data called Similarity-based Formal Concept Analysis (SFCA) [7] are used as data mining techniques for this purpose. This module contains three main components:

1. the Context Builder is responsible for preparing the input data-set to the classification module. It selects the main properties of Web Services and creates a tabular representation where the rows correspond to the Web Services, the columns correspond to the services capabilities (descriptions) such as type of input or the ontology that the input refers to, and finally table cells contain values of these properties for each service.

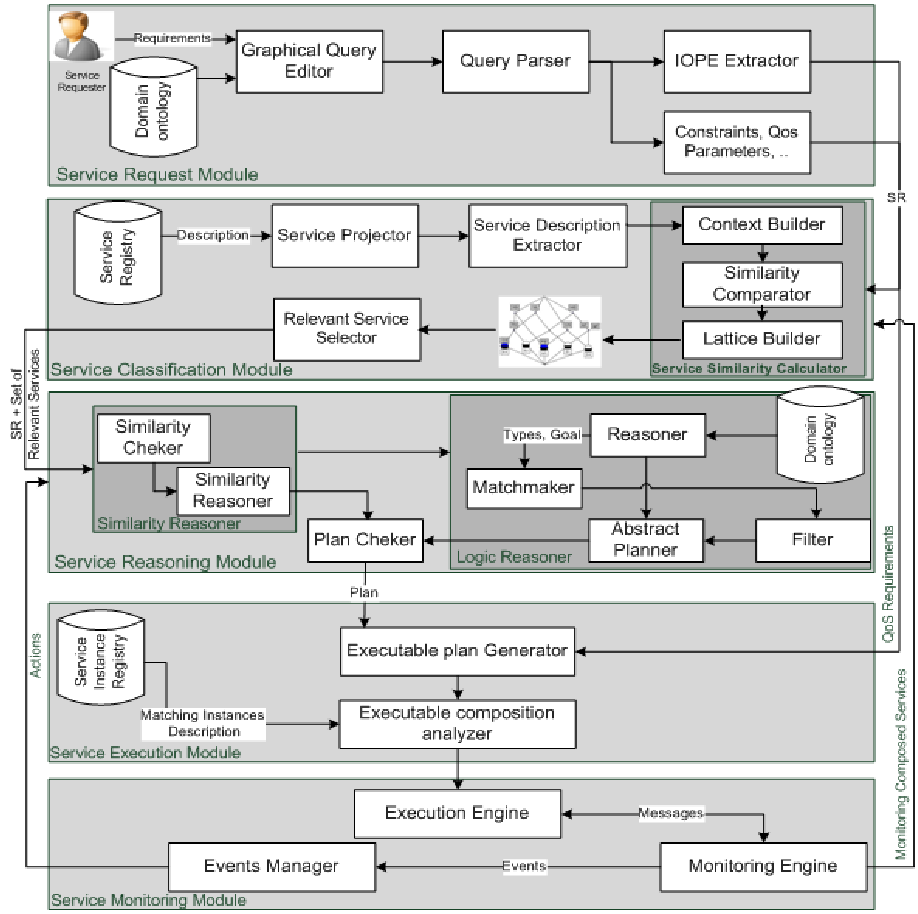


Fig. 2. IDECSE Architecture

2. the Similarity Comparator relies on a set of mathematical formulas based on the semantic distance between concepts from the Context Builder and then between services in the registry.
3. the Lattice Builder enables to compute the lattice structure corresponding to the input table generated by the Context Builder. The lattice structure reflects services grouping possibilities based on their common or similar properties.

Once the lattice is built, services are grouped into classes according to their similarities. The Relevant Service Selector identifies the most relevant classes of services from the lattice (lattice interpreter), which is more likely to answer the query. The set of relevant services can then be outputted with the appropriate rank with respect to the query needs (Service Ranking) to the next module.

2.3 Service Reasoning Module

The Service Reasoning module identifies the candidate composition plans that realize the goal through a similarity-based reasoner or a logic-based one. The Similarity Reasoner is based on the value of semantic similarity calculated by Service Similarity Comparator module. When the semantic similarity value between S_R and one or more existing services in the registry is higher than a given threshold then S_R is satisfied. Otherwise the Logic Reasoner is called to identify the plan of services that achieves the goal of S_R. The main components of the Logic Reasoner module are:

- Reasoner: checks the ontology consistency in addition to handling the maintenance of state including preconditions evaluation and effects application.
- Filter: avoids redundancy from the plan by identifying service types with potential relevance to the goal and checks the dependency relationships between each two consecutive service types.
- Matchmaker: allows querying the service registry for available services in order to match the preconditions of a Web service with the effects of another.
- Abstract Planner: It can be considered as the main component of this module and is responsible for generating a set of abstract plans.

To determine an Abstract Plan, the composition is reduced to a planning problem. A Plan is formalized as a proof of the goal to answer the user query. A Plan $\mathcal{P} = \{A_i\}_{i=1..n}$ is a sequence of n actions. Each action A_i applies on a state E_i to produce a state E_{i+1}: $\forall i \in \{0, .., n-1\}, E_i \wedge A_i \models E_{i+1}$. Starting from the initial state E_0 the plan $\mathcal{P}$ produces the goal G: $E_0 \wedge \mathcal{P} \models G$.

2.4 Service Execution Module

The service Execution Module translates the abstract plan into an executable one by associating to each service type its specific instances using the Service Instances Registry. The plan generated by the logic reasoner is considered as a template for the composite service and drives the process of matching each service type to a corresponding service instance. The Service Execution Module is mainly composed of two main components: The Executable Plan Generator considers non-functional requirements of the goal and enables to concretize the abstract plan generated by the Service Reasoning Module. The Executable Composition Analyzer generates executable code and invokes the execution engine in the Service Monitoring module. Different works was proposed in order to implement the Executable Plan Generator, we can use for example the algorithm presented in [8]. This algorithm takes as input a composition plan, the QoS permissible values imposed by the user, and their weights and generates as output a composition plan that satisfies the requirements of the user.

2.5 Service Monitoring Module

Monitoring deals with the actual execution of the composite service and is responsible for monitoring the execution and recording violation of any requirement of the goal service at runtime. If a violation event occurs, an adaptation engine is triggered to handle this violation. There are two inputs to a monitoring framework, a set of constraints that the process must obey and events or messages generated during the execution. A processing engine ensures that all events comply with the constraints and reports any exception.

3 Related Works

A considerable number of research efforts have focused on various aspects of Web service compositions ranging from semantic service discovery to semantic specification, deployment, and monitoring. MoSCoE (Modeling Web Service Composition and Execution) [3] aims to provide a model-driven framework for an automatically composition of services. MoSCoE allows service providers to publish their services in a standard and semantic way and uses UML state machines for visually representing composite services. The Composition process in MoSCoE is based on the three steps of abstraction, composition and refinement. However user preferences and QoS were not addressed but only outlined as future work. In [9], authors combine semantic service descriptions with the invocations of the WSDL descriptions allowing to execute the composed services on the Web. The process includes matching services to the user at each step of a composition and filtering the possibilities by using semantic descriptions of the services. The generated composition is then directly executable through the WSDL grounding of the services. [10] presents a framework for service composition based on functional aspects, in which services are chained according to their functional and semantic description (IOPEs). The proposed framework uses the Causal Link Matrix (CLM) formalism in order to facilitate the computation of the final service composition as a semantic graph. The set of possible solutions are then pruned, at composition time, in order to rank the service compositions according to some fixed criteria. These criteria can be defined based on the semantic similarity of component services and/or the non-functional properties of the compositions calculated by aggregating the non-functional properties of the component services. In [11] and [12], Description Logics (DL) frameworks for Semantic Web service composition are proposed. The specification of Semantic Web services is reduced to preconditions, which define logical conditions that should be satisfied prior to the service invocation and effects, which are the result of the execution of the service. The proposed approaches for services composition use backward chaining search algorithms to find potential candidate services. Thus the composition process is done automatically and dynamically.

IDECSE builds on the approaches mentioned above and aims to provide a declarative approach to service composition engineering to achieve a full manipulation of the composition process. IDECSE appeals for data mining techniques for classifying and mining services into service registry in order to anticipate relevant services search and reduce services search space. It also considers monitoring and adaptation concerns, which are not incorporated in the above approaches. IDECSE provides an easy way to specify functional and non-functional requirements of composite services in a semantic and declarative manner, guides the user through the composition process, while allowing modifications or feedbacks.

4 Conclusion

This paper describes IDECSE, a new semantic integrated approach for composite services engineering. Compared to existing approaches IDECSE considers semantics in all the composition global life-cycle, addresses the challenge of fully automating the composition processes, uses data mining techniques such as SFCA for classifying and mining services, and proposes new reasoning, monitoring, and adaptation techniques. Our work in progress includes the improvement and implementation of the different modules of the proposed architecture. We also plan to extend the framework to include additional features such as failure handling, and an interactive visual environment for testing composite services.

References

1. Batra, V.S., Batra, N.: Improving Web service QoS for wireless pervasive devices. In: 2005 IEEE International Conference on Web Services, IEEE (2005) 130–137
2. Bener, A.B., Ozadali, V., Ilhan, E.S.: Semantic matchmaker with precondition and effect matching using SWRL. Expert Systems with Applications (2009) 9371–9377
3. Pathak, J., Basu, S., Lutz, R., Honavar, V.: MoSCoE: A framework for modeling Web service composition and execution. In: Data Engineering Workshops, 2006., IEEE (2006)
4. Kona, S., Bansal, A., Gupta, G., Hite, D.: Automatic composition of semantic Web services. In: ICWS 07. (2007) 150–158
5. Rao, J., Su, X.: A survey of automated Web service composition methods. In: Semantic Web Services and Web Process Composition. Springer (2005) 43–54
6. Ganter, B., Wille, R.: Formal Concept Analysis: mathematical foundations. 1999
7. Azmeh, Z., Hamoui, F., Huchard, M., Messai, N., Tibermacine, C., Urtado, C., Vauttier, S.: Backing composite Web services using Formal Concept Analysis. In: Formal Concept Analysis. Springer (2011) 26–41
8. Ko, J.M., Kim, C.O., Kwon, I.H.: Quality-of-service oriented Web service composition algorithm and planning architecture. Journal of Systems and Software (11) (2008) 2079–2090
9. Sirin, E., Hendler, J., Parsia, B.: Semi-automatic composition of Web services using semantic descriptions. WSMAI (2003) 17–24
10. Lécué, F., Silva, E., Pires, L.F.: A framework for dynamic Web services composition. In: Emerging Web Services Technology, Volume II. Springer (2008) 59–75
11. Qiu, L., Lin, F., Wan, C., Shi, Z.: Semantic Web services composition using AI planning of Description Logics. In: IEEE Asia-Pacific Conference on Services Computing, 2006, IEEE (2006) 340–347
12. Wang, Y.: Research on Web service composition algorithm using Description Logics. TELKOMNIKA Indonesian Journal of Electrical Engineering (1) (2014) 852–858

Specification and Implementation of a Meta-model for Information Systems Cartography

Gorica Tapandjieva and Alain Wegmann

École Polytechnique Fédérale de Lausanne,
Systemic Modeling Laboratory LAMS
Station 14, CH-1015 Lausanne, Switzerland
{gorica.tapandjieva,alain.wegmann}@epfl.ch,
WWW home page: http://lams.epfl.ch/

Abstract. Models are used in software engineering, enterprise architecture, requirements engineering, etc. In this context, models can give support in creating a shared understanding of what exists in an enterprise. For this purpose we suggest to use a cartography tool that stores and displays the variety of models. But cartography tools have limitations in the number of enterprise levels shown. Since every model has to conform to a unique meta-model, in this paper we report on a development of a SEAM meta-model used within the Solu-QIQ cartography tool to overcome the limitations of the default cartography meta-models.

Key words: meta-model, SEAM, URBA, enterprise architecture, EA tool, Solu-QIQ

1 Introduction

Enterprise architecture (EA) is defined as "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure" [8]. Today there are many EA frameworks and methodologies that differ in their approaches and in their level of details. EA comparison studies can assist architects in choosing the appropriate framework or methodology [12, 15]. In this paper we look at EA diagrams as an information systems (IS) cartography.

Having a cartography in an enterprise is important because it helps enterprise architects, managers, governance bodies and all employees in general, to create a shared understanding of what exists within and around the enterprise. With this, decisions that optimize the usage of Information Technology (IT) resources are made more easily. In the literature, an information system cartography is also known as a top-down urbanization approach (IS Urbanization and Enterprise Architecture, URBA-EA) [10], and it comes with the possibility to use a cartography tool.

Any EA tool which outputs a complete EA landscape and which offers a fast way of EA modeling without losing the models correctness can be seen as cartography tool. We decouple the tool from the approach implemented in the tool. As a consequence, in Section 2, we give an overview of few EA tools and EA approaches separately. Section 2 also contains matrices comparing the tools and approaches according to features we find important.

In Section 3 we describe the chosen approach and its meta-model. In Section 4 we show how this meta-model is implemented in the chosen tool. Finally, we give conclusions and present the future work.

2 EA Tools and Approaches

People creating and using models need a modeling tool that speeds up their work. At the same time, their models must not lose the correctness. When we were looking for a tool to do the modeling, we focused on the following criteria:

A. Ability for meta-model customization, making the tool decoupled from it's integrated EA approach.
B. Mass-modeling capability and various formats of data and model import/export.
C. Automatic diagram layout generation.

Our evaluation of some tools is presented in Table 1.

	A	B	C
Iteraplan [6]	No	Yes	Yes
Enterprise Architect [4]	No	Partial	No
Solu-QIQ [3]	Yes	Yes	Yes

Table 1. Summary of some tools evaluation criteria

As for tools, we had several criteria for the EA modeling approach. We looked at frameworks, modeling languages and methodologies, and the questions we asked are:

I. Does the approach have a notation?
II. How many levels (domains) does the approach model?
III. Is the approach's meta-model generic between different levels or domains?
IV. Is the approach declarative?

Our review of some approaches is given in Table 2.

Based on our criteria, we choose Solu-QIQ tool with the SEAM approach. With SEAM's ability to model multiple levels, we model the enterprise from the business down to IT, using the same notation. Also, the declarative way of modeling is an advantage when focusing on macro scenario descriptions. Solu-QIQ makes this approach more powerful with it's mass modeling and automatic diagram-generation features.

	I	II	III	IV
TOGAF [14]	No	4	No	No
ArchiMate [1]	Yes	4	No	No
URBA [10]	No	4	No	Depends on the level
SEAM [5]	Yes	User defined	Yes	Yes

Table 2. Summary of some EA approaches evaluation criteria

3 Systemic Enterprise Architecture Methodology – SEAM

SEAM is a family of methods used for consulting and teaching strategic thinking, business/IT alignment, and requirements engineering [7, 16]. In this paper we elaborate on applying SEAM in the creation of an IS cartography, so we use the Enterprise Architecture specific SEAM.

SEAM represents an organization as a **hierarchy of service systems** (from business down to IT, also known as organizational level hierarchy). To clarify, we use **system** to refer to any kind of entity in our surrounding: an organization, an employee, an IT system, or an application [17]. In General Systems Thinking (GST), the common definition of a system is "a set of elements standing in interrelations"[18] and every system (and service system) has a boundary.

(a) Systems (b) Actions (c) Refinement (decomposition) and invoke (use) link respectively

Fig. 1. SEAM notation for the different kinds of service systems, actions and links

In the mentioned hierarchy, we can model systems as a **whole, also denoted as [w] (black boxes)** or as a **composite, denoted as [c] (white boxes)**. By modeling a system as a whole, we ignore the system's components and we focus only on the services offered by the system. When we model a system as a composite, the components and their relationships are visible, so we see the implementation of the service and understand the responsibility of each component. In this case, the system seen as a composite is the parent in the organizational level hierarchy and the component systems are placed within the boundary of their parent. Essentially, there are no rules on the number of levels in the hierarchy.

Besides the organizational level hierarchy, we can specify the behavior (functionality) of each system. There are two kinds of functionalities [17]:

1. **Service (localized action)** is the behavior of a system as a whole. It represents *a service offered by a system.*
2. **Process (action binding)** is the behavior of a system as a composite and it represents *a service implementation by a system* [2]. Usually the process is connected to services from other systems as a whole (that share the same system as a composite parent).

When we have both of the views (as a whole and as a composite) for the same system in one diagram, we can observe two types of connections (links), depicted in Fig. 1(c):

– The two views are connected with a *refinement* (decomposition) link. Essentially, this link shows that it is the same system. The process in the system as a whole corresponds to the implementation of the service in the system as a composite.
– In the system as a composite, the process is connected with plain links to other services that belong to the component systems as a whole. These links mean that the process invokes (uses) those services.

As a notation, a block arrow pictogram is used to specify a general (business) service system in a SEAM diagram. There exist other pictograms that specify the nature (type) of the service system (see Fig. 1(a)). The name of the system is written on the top of the pictogram, followed by a small letter in square brackets [w] or [c], denoting if it is a system as a whole or a composite respectively. Also, the pictograms used for a service and a process are depicted in Fig. 1(b).

An example of a SEAM model is depicted in Fig. 2.

In a service oriented environment, there is no definite number of layers between the application layer and the layer where the final end user participates in/consumes the service. In Fig. 2 we show four organizational levels in a SEAM model. In this example model there is an application in the third (*Application 22*), and in the fourth level (*Application 31*). We are not showing details of the application and infrastructure layers. They can be seen after we model its composite views, but for simplicity we don't show them here.

If we try to map this model to a meta-model with fixed number of levels, we will fail. The application layer is usually fixed in one level, and we model a situation where we need an application layer in two different levels. Adding an extra application layer can solve the problem, but it also adds complexity and decreases scalability of the overall solution.

Another solution to the problem described is defining the levels to be recursive, which is possible with SEAM. If the information about the level is needed, it is possible to store it as a flag.

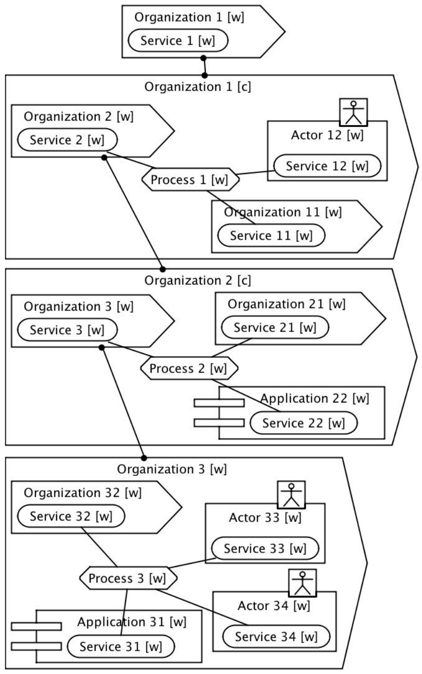


Fig. 2. Example SEAM model showing the organizational level hierarchy

4 SEAM Meta-model Implementation in Solu-QIQ

SEAM is capable of modeling more scenarios compared to other approaches, which is a great advantage. But this advantage makes it difficult to find a meta-model that every SEAM model will follow. Finding this meta-model enables modelers to create SEAM models that capture more situations in a fast way by using the Solu-QIQ tool.

The already developed meta-models, which capture all methods which are part of SEAM [11, 9] are very exhaustive. From the SEAM basic EA method explanation in Subsection 3, the meta-model should capture systems (whole and composite), actions (service and process) and connections (decomposition and usage). We also derive several modeling rules, that help in making the meta-model more simple. We summarize the rules as follows:

- A system as a whole hosts only services, and a system as a composite hosts processes and other systems as a whole.
- A process is linked with (uses) services from systems as a whole.
- The decomposition link, connects different views of a system.

From this we conclude that the only items that should be present in our meta-model are: **a system, a service** and **a process**. Our proposed meta-model is shown in Fig. 3.

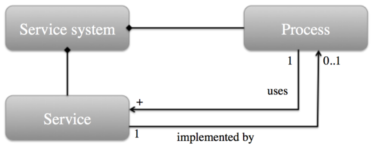


Fig. 3. Meta-model of SEAM concepts implemented in Solu-QIQ

In this class diagram of the meta-model we see two *Composition* links (**Service System** – **Process** and **Service System** – **Service**). By these two links we have partly captured the system *decomposition link*, and the *whole* and *composite* concepts. Let's say we have a service system **sysX**, that offers the service **serX** (when **sysX** is seen as a whole), and a process **procX** that implements the **serX**, which is in **sysX**, when the system is seen as a composite. Whenever a system is connected with a service, we know that we are looking at its view as a whole. The same holds for a system connected with a process and a system's view as a composite. This decomposition scenario will be complete only when we know that **procX implements serX**. This is done by the connection between Service and Process in the meta-model.

Finally, we have a many-to-many relationship between a Process and a Service. The reason for this is the invoke (use) link. The Service → Process (decomposition) link stands for "the service implemented by a process", and the Process → Service (invoke) link stands for "the process uses a service". The cardinality of processes and services is different. On the process side it is 0..* because there can be services for which we don't know the implementation. For the services it is 1..* because there can not be a process that doesn't use a service. Also, once we have a process, we have the corresponding service. The distinction between these two types of links is stored as a flag in this many-to-many relationship.

The Service System in the meta-model has one additional attribute - Type, of the SystemType. This SystemType is a typology that is used to distinguish what kind of system we want to show (see Fig. 1(a)). With this we implement the solution of having recursive layers. By giving the value *Application* to the system's type, we will know that that system is an application and belongs to the corresponding Application layer in URBA.

We tested the proposed meta-model by implementing it in Solu-QIQ. It is not intuitive that this meta-model is capable of showing infinite number of organizational hierarchy levels. For this purpose, we populated the meta-model in the tool with data that correspond to the SEAM model seen in Fig. 2. It is not enough just to insert data. It is up to us how do we interpret the meta-model and the data present. This is done by creating a query in Solu-QIQ that outputs enough data to show one organizational hierarchy level as a SEAM model. The final Solu-QIQ output is depicted in Fig. 4 and it is obtained after defining a style in the tool for the output of the query. We have to note that the output shows only one level at a time, and in this figure four separate outputs are shown.

When we compare Fig. 2 and Fig. 4, we can see that it is the same model. With this we proved that Solu-QIQ is able to generate SEAM models based on

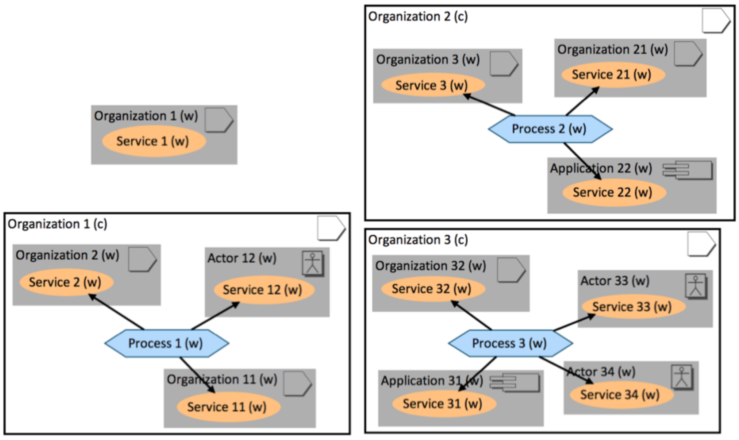


Fig. 4. A Solu-QIQ output of the SEAM model shown in Fig. 2 implemented based on the meta-model shown in Fig. 3

the meta-model we provided. So in future, instead of drawing SEAM models, system by system, action by action, we can only populate the Solu-QIQ tool with the needed data, click on a button, and we will have the model ready.

5 Conclusions

In this paper we show how to combine an EA approach with a tool independent of the approach through creating a mutually compliant meta-model. This tool is further on used for generating EA models. The overall goal is to have models of the EA that enables the IT to give better support to the enterprise's IT strategy. This is tightly connected with the definition and implementation of an IT strategy, as seen in [13].

In this paper we describe the development of a meta-model that is compliant with our chosen EA approach, SEAM, and an EA tool, called Solu-QIQ. This tool is inspired by the urbanization approach.

First, we explained why SEAM and a cartography tool such as Solu-QIQ are needed in an enterprise, and why it is important for them to coexist. Then, we gave a simple meta-model that captures the basic SEAM principles. We finish with the implementation of this meta-model in Solu-QIQ and show the output of a generated model. By doing all this, we showed the recursive side of SEAM and we display the efficiency of Solu-QIQ when dealing with recursive meta-models.

6 Future Work

Our next step is adding a concept in the meta-model that shows grouping of services by functionality. As SEAM is a service-oriented approach, this grouping will also give the service catalog.

The presented meta-model captures only the basic SEAM usage and capability. SEAM is much more powerful than showing services, processes, service systems as a whole and as a composite. In order to benefit from the complete SEAM approach, this meta-model has to be augmented with the other SEAM concepts not presented here.

References

1. ArchiMate, an Open Group Standard. http://www.opengroup.org/subjectareas/enterprise/archimate.
2. B. Bajić-Bizumić, C. Petitpierre, H. C. Huynh, and A. Wegmann. A model-driven environment for service design, simulation and prototyping. In *Exploring Services Science*, pages 200–214. Springer, 2013.
3. BeLINK. http://www.groupe-belink.fr/.
4. Enterprise Architect. http://www.sparxsystems.com/products/ea/.
5. A. W. et al. Requirements Modeling in SEAM: The Example of a Car Crash Management System. In *Comparing Requirements Modeling Approaches (CMA@RE) workshop*, 2013.
6. Iteraplan. http://www.iteraplan.de/en/eam-iteraplan.
7. LAMS. Seam for Teaching. http://lams.epfl.ch/reference/seam/.
8. M. Lankhorst. *Enterprise Architecture at Work: Modelling, Communication and Analysis*. Springer, 2009.
9. L.-S. Lê and A. Wegmann. Hierarchy-oriented modeling of enterprise architecture using reference-model of open distributed processing. *Computer Standards & Interfaces*, 35(3):277–293, 2013.
10. C. Longépé. *The Enterprise Architecture IT Project: the Urbanisation Paradigm*. Butterworth-Heinemann, 2003.
11. I. Rychkova. Formal semantics for refinement verification of entreprise models. *Unpublished doctoral dissertation, University of Lausanne, Switzerland*, 2008.
12. R. Sessions. Comparison of the Top Four Enterprise Architecture Methodologies, 2007.
13. G. Tapandjieva, D. R. Marchetti, I. Rychkova, and A. Wegmann. Towards the Definition, Implementation and Communication of an IT Strategy: the Case of IT Strategy at EPFL. In *The 8th International Workshop on Business/IT-Alignment and Interoperability*, 2013.
14. The Open Group Architecture Framework (TOGAF). http://www.opengroup.org/subjectareas/enterprise/togaf/.
15. L. Urbaczewski and S. Mrdalj. A Comparison of Enterprise Architecture Frameworks. *Issues in Information Systems*, 7(2):18–23, 2006.
16. A. Wegmann. On the Systemic Enterprise Architecture Methodology (SEAM). In *Published at the International Conference on Enterprise Information Systems 2003 (ICEIS 2003)*. Citeseer, 2003.
17. A. Wegmann, A. Kotsalainen, L. Matthey, G. Regev, and A. Giannattasio. Augmenting the Zachman Enterprise Architecture Framework With a Systemic Conceptualization. In *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE*, pages 3–13. IEEE, 2008.
18. G. M. Weinberg and J. Wiley. *An Introduction to General Systems Thinking*. Wiley New York, 1975.

Anticipation-driven Architecture for Proactive Enterprise Decision Making

Babis Magoutas[1], Nenad Stojanovic[2], Alexandros Bousdekis[1], Dimitris Apostolou[1], Gregoris Mentzas[1], Ljiljana Stojanovic[2]

[1]Information Management Unit, National Technical University of Athens, 9 Iroon Polytechniou str., 157 80 Zografou, Athens, Greece
{elbabmag, albous, dapost, gmentzas}@mail.ntua.gr

[2] FZI Research Center for Information Technologies, Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Germany
{nstojano, stojanov]@fzi.de

Abstract. In this paper we present a visionary approach about a new architecture for supporting proactive decision making in enterprises. We argue that a cognitive approach of continuous situation awareness can enable capabilities of proactive enterprise intelligence and propose a conceptual architecture outlining the main conceptual blocks and their role in the realization of the proactive enterprise. The presented approach provides the technological foundation and can be taken as a blueprint for the further development of a reference architecture for proactive enterprise applications. We illustrate how the proposed architecture supports decision-making ahead of time on the basis of real-time observations and anticipation of future undesired events by presenting a practical application in the oil and gas industry.

Keywords: proactivity, predictive analytics, enterprise, decision-making, event-driven computing, condition-based maintenance.

1 Introduction & Motivation

Today's enterprises are facing increasing pressure due to globalization, uncertainties, and increased regulations, among others. These pressures are forcing the companies to manage production at the margins of performance, achieving better control through the whole of the production process. As an example companies need to know what goods are in transit, what is about to enter the warehouse, what is being shipped from suppliers, in order to dynamically route goods in-transit.

To cope with these challenges in general, dynamic business networks need to enhance their monitoring capabilities within the network and across different levels. To achieve this, real time monitoring can be used and such kind of real-time monitoring have been already implemented in many enterprises in the form of extensive sensor/IoT- systems. Indeed, sensing enterprises are a reality, starting from manufacturers that can sense some deviations from the production plan as soon as they appear, till large logistics networks that sense delays about the delivery time in real-time. The main driving concept in sensing enterprises is events and correspondingly, the event-driven architecture (EDA) is underlying their realization.

However, event monitoring, is only the first, crucial step to manage problems in complex, dynamic systems. Next step is enabling that event monitoring copes with the scale and dynamics of the business context (internal and external). Indeed, change is constant, therefore monitoring solutions must also change so they can adapt and stay relevant. For example, changes in the business performances should be registered as soon as they happen and taken as new monitoring goals.

This kind of dynamic monitoring is the basis for the new level of (sensing) performance observing that is not only sensing the problems, but also sensing that the problem might appear, i.e. focusing on a proactive approach. Indeed, observing a delay is very useful information, but anticipating that there will be a delay is far more important from the business point of view. Moreover, such anticipation will lead to the possibility to act ahead of time, i.e. to be proactive in resolving problems before they appear or realizing opportunities before they become evident for the entire business community and be able to recover and support continuity. This ability to support continuity in operations at the margins is called resilience [1], and is a key strategy in today's and future industrial operation. From the architecture point of view this requires reorientation from events as changes that happened in time to anticipation as prediction that something will happen in near future.

In this paper we present a visionary approach about anticipation-driven sensing and decision-making that will enable the transition from sensing to proactive enterprise. One of the main novelties is the treatment of anticipation as the first class citizens in our approach: it supports the whole life-cycle of the anticipation, from sensing/generating anticipations till validating the reactions (proactions) based on them. We argue that a cognitive approach of continuous situation awareness can enable capabilities of proactive enterprise intelligence and propose a conceptual architecture for proactive enterprises systems. We present an application scenario for proactive decision-making in the area of condition-based maintenance.

The rest of the paper is organized as follows. Section 2 discusses our vision for the proactive enterprise, while Section 3 outlines the proposed approach for realizing such a vision. Section 4 presents our proposed architecture and Section 5 an envisaged scenario where proactivity can be injected in enterprise decision-making. Section 6 discussed related work, while Section 7 concludes the paper.

2 Vision

Our vision for the proactive enterprise compared to the current reality of sensing enterprise is illustrated in Figure 1. Sensing enterprises are operating on the surface of the possibilities (the tip of the iceberg), whereas a deeper diving into the endless wealth of opportunities is required in order to enable the transition to the proactive enterprise. Consequently, like the events are driving reactivity in the sensing enterprise, anticipations (predictions) are driving proactivity in proactive enterprise leading to increased situation awareness capabilities even ahead of time (cf. Figure 1). This requires new methods and technologies that are responsible for dealing with anticipations, which are part of the novel anticipation-driven architecture:

- Anticipation based on Big data - Exploiting the power of big enterprise data, by sensing the whole business ecosystem: shifting relevant business context from internal processes to the ecosystem
- Anticipation-based Actions - Extracting the actionable meaning from data, by applying advanced big data predictive analytics: shifting the processing capabilities from real-time into ahead-of-time processing
- Anticipation-driven Optimization - Increasing the strategic value of data analysis for decision making, by dynamically adapting patterns of interest found in real-time big data streams and enabling proactive decisions: shifting decision making focus from early warnings into business optimization

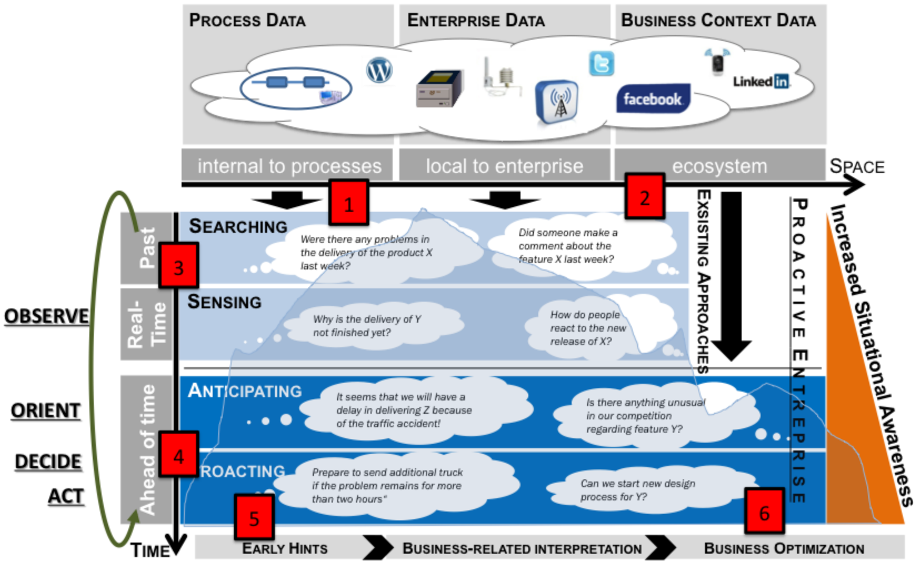


Figure 1: From Sensing till Proacting. **(1-> 2, Space axis: From Processes to the Business Ecosystem; 3-> 4, Time axis: From Real-time to Ahead of Time Processing; 5 -> 6: Decision Making: From Early Hints to Business Optimization)**

This will lead to a new class of enterprise systems, proactive and resilient enterprises, that will be continuously aware of that what „might happen" in the relevant business context and optimize their behavior to achieve what "should be the best action" even during stress and balancing on demanding margins. Proactive enterprise systems will be able to suggest early on to the decision makers the most appropriate process adjustments to avoid singular system behavior and optimize its performance.

This paper introduces a novel architecture for realizing proactive enterprise, encompassing anticipations as the first class citizens and driver of the processing. The architecture is based on the Observe, Orient, Decide, Act (OODA) loop of situational awareness that has been recognized as one of main models for the big data supply chain [2] - the key for continuous situational awareness. This model sees decision-making occurring in a recurring cycle of unfolding interaction with the environment, oriented via cues inherent in tradition, experience and analysis. These cues inform hypotheses about the current and emerging situation that, in turn, drive actions that test hypotheses.

The first step towards continuous situation-aware proactivity is to enable comprehensive observing of the relevant enterprise context/ecosystem through the design and development of a smart sensing system able to cope with a huge amount of heterogeneous (big data) in real-time, focusing on predictive sensing (sensing early warnings – anticipations). Next, semantic understanding of acquired data in near real time should be enabled (orient) by designing and developing an efficient management framework for dynamic (proactive) and context-aware anticipation and detection of the situations of interest on the basis of complex and predictive data analysis algorithms and event-detection. This will provide the basis for supporting making decisions and actions ahead of time through designing and developing mechanisms for the proactive recommendations based on the dynamic situational awareness and the predictive data analysis. An example of this dynamic mechanism is the use of proactive indicators to support resilience. Finally, proactive handling that will result in sustainable business improvements should be ensured, through designing and developing methods for defining and dynamic monitoring of KPIs and corresponding adaptation of the whole OODA cycle, closing the feedback loop and leading to the continual proactive business optimization. Figure 1 illustrates how our approach for realizing the OODA loop can be seen in the context of the proactive enterprise vision.

3 Conceptual Architecture

Based on the proposed continuous situational awareness approach presented above, we outline the main conceptual blocks and their roles in the realization of the proactive enterprise platform, an anticipation-based platform for integrating heterogeneous real-time and dynamic streams created by hardware sensors, software and external data used in enterprises. The proposed conceptual architecture is strongly oriented on the OODA loop and combines services of smart sensing, anticipation management, incremental proactivity and proactivity management (see Figure 2).

Smart sensing services include adapters, pub-sub middleware and the Scalable Event Storage. Adapters enable communication with all necessary enterprise information sources such as hardware sensors (which might include vibration and temperature sensors, environmental sensors), software sensors from ERP and other enterprise systems and external business context data. Pub-sub middleware is realized as an event cloud, a scalable, P2P based repository that delivers RDF events to the requesting parties (subscribers) and ensures the decoupling between components so that the system can scale very easily. The Scalable Event Storage enables semantic enrichment with background knowledge of real-time streams and allows storage of events (in the form of RDF triplets) received from adapters for historical and statistical purposes. It supports synchronous and asynchronous queries expressed in a subset of the SPARQL language and accessible through corresponding APIs.

Services for anticipation management will enable the generation of real-time, data-driven predictions, as well as the discovery of unusual situations, based on events delivered by storage. Novel predictive analytics services will be realized as intelligent services on the top of probabilistic stream processing technologies. The Complex Event Processing (CEP) component has the role of dynamic definition and detection of complex events and reasoning over events, supplied by Event Storage. Complex Event Patterns can be defined and deployed dynamically or produced by offline analytics. CEP allows goal-driven identification of relevant situations of interest and leverages detection of anomalies in real-time providing the basis for proactive actions.

Incremental proactivity service subscribes for predicted situations of interest (pub/sub communication) and generates corresponding proactive recommendations, by taking into account business context. It couples dynamic and uncertain decision making methods and decision theoretic optimization models and proactively recommends actions and activation time maximizing the utility for the enterprise, while considering several criteria such as cost, time and safety.

Proactivity Management deals with defining and dynamic monitoring of KPIs and corresponding adaptation of the whole OODA cycle, closing the feedback loop and leading to the continual proactive business optimization.

By taking into account the complexity of the business environment the modern enterprise is working in (Big Data, Dynamic Context, Critical Decision Making), we argue that this architecture, by assuming that it will be validated through use cases, can be taken as a blueprint for further development of a reference architecture for Proactive enterprises.

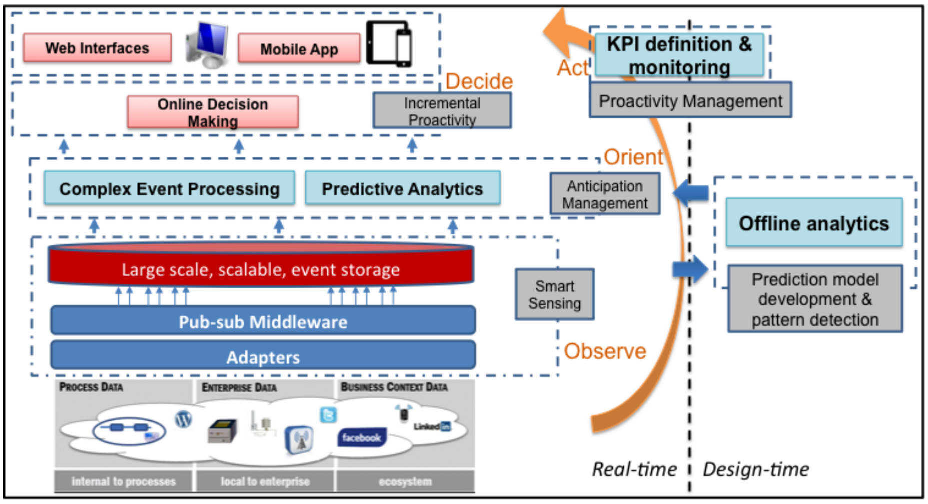


Figure 2: Conceptual Architecture

4 Envisaged Scenario & System Walkthrough

In this section we present a practical application of the proposed framework for anticipation-driven sensing and proactive decision-making, in the oil and gas industry. We describe the practical role and use of the proposed framework focusing on how it can support decision-making ahead of time on the basis of real-time observations, predictions and anticipation of future undesired events, through an indicative scenario of proactive condition-based maintenance (CBM).

CBM in the oil and gas industry employs various monitoring means to detect deterioration and failure in some critical drilling equipment. Equipment failure situations can be forecasted/anticipated based on observations of events related to this equipment or the surrounding environment; e.g monitoring engine temperature indicators, monitoring electric indicators (measuring change in the engine's electric properties) and performing oil analysis [3]. In reality, several different patterns will imply various failure distributions; for the sake of the example, we will focus on thermal indicators, adopting the approach presented in [4]. More specifically, in this scenario we focus on the gearbox drilling equipment and consider as indicators the rotation speed of the drilling machine's main shaft in RPM, along with the lube oil temperature of the drilling machine's gearbox.

We distinguish between two different types of operations performed by components of the proposed framework in order to support anticipation management; offline and online operations performed at design time and real-time, respectively. At design-time the **offline analytics** component extracts from historical data of oil temperature, RPM events and gearbox equipment failure, the distributions associated with gearbox breakdown along with their relation to monitored indicators and builds a breakdown prediction model that will enable the generation/detection of the anticipation of interests. Moreover, it identifies which complex event patterns indicate that a drilling gearbox equipment failure starts to occur, on the basis of historical data, and communicates the identified pattern to CEP for real time monitoring.

At real-time, the **CEP** component detects a complex pattern of simple oil temperature and RPM events characterized by an abnormal oil temperature rise (10% above normal) measured over 30% of the drilling period when drilling RPM exceeds a threshold, caused by abnormal friction losses in the drilling gearbox during drilling. This pattern, learned at the offline phase, is a strong indication that a gearbox equipment failure starts to occur. Based on the pattern detected by CEP in real-time, the **online predictive analytics** component analyzes the current trend of oil temperature and RPM increases and drops for the most recent events and predicts (anticipates) the occurrence of a future gearbox break down along with the associated probability distribution function, based on trend analysis and the breakdown prediction model learned at the offline phase.

Based on the predicted probability distribution for the occurrence of a future gearbox breakdown, the **online decision-making** component provides proactive recommendations of actions that either mitigate (i.e. reduce the probability of occurrence) or completely eliminate the future gearbox breakdown, along with the recommended activation time. This component applies dynamic and uncertain decision making methods and decision theoretic optimization models that minimize cost or maximize the utility for the oil drilling company. Examples of actions aiming to optimize the maintenance policy according to cost criteria may be a) to take the equipment down for full maintenance - an action that completely eliminates the predicted gearbox breakdown - or perform less costly actions that only reduce the probability of failure such as b) perform lubrication of metal parts, or c) shift drilling to lower pressure mode. Actions could also be related to resource management and organization of the resources needed to rectify the gearbox failure in case it occurs.

The business added value of anticipation-driven proactive decision-making in this scenario is huge. With a typical day rate for a modern oil rig being around USD 500 000, reducing undesired downtime, with its associated high cost (one hour of saved downtime is typically worth USD 20 000) is of outmost importance in the oil drilling industry. Therefore, we expect that the proposed framework, which is able to provide early notifications about equipment problems and proactive recommendations about optimal decisions on the basis of utility, cost and other factors, will allow proactive enterprises in the oil and gas industry to gain a strong competitive advantage based on reduced downtimes and optimized performance.

5 Related Work

Although the idea of proactive computing may seem simple, the quantity and quality of proactive applications is rather modest. Proactive applications have been developed in an ad-hoc manner for several years; applications regarding proactive decision-making include network management [5], supply chain management [6] scheduling of manufacturing systems [7] and maintenance [8].

Especially maintenance has gathered significant research interest. Although there is not a complete agreement in the literature about the classification of maintenance types, they can generally be divided to three categories: breakdown maintenance which takes places when a failure occurs, time-based preventive maintenance which sets certain activities when a defined period of time passes and Condition Based Maintenance (CBM) which recommends actions according to the health state of the manufacturing system [9]. In CBM, real-time proactive decision support becomes significant because a maintenance strategy, usually based on a prognosis model, needs to be implemented [8]. Several techniques have been developed within the framework of CBM by utilizing OR, AI, multiple criteria methods and several statistical techniques accompanied with the appropriate architecture [10].

Despite these applications, the lack of a generic paradigm to develop proactive event-driven applications makes it difficult for this capability to spread. Because of its nature, proactive computing requires an integration of various technologies for sensing, real-time processing and decision-making. The approach presented in this paper provides the technological foundation and can be taken as a blueprint for the further development of a reference architecture for proactive enterprise applications. With respect to the maintenance domain, which is the application domain of the presented envisaged scenario, our approach goes beyond time-based preventive maintenance by extending stochastic preventive maintenance methods [11] and integrating them in an innovative anticipation-driven ICT architecture.

6 Conclusions and Future Work

We argue that the proposed architecture for anticipation-driven decision-making is the ultimate basis for realizing proactive enterprise. This has several implications for practitioners. They need to be prepared both in technical terms and from a cognitive perspective to take advantage of the novel business intelligence capabilities that will be provided. On the one hand they need to design and implement physical (such as smart sensors and actuators, location-aware sensors, cyber-physical systems) and virtual sensors (such as agents in customer transaction and relationship systems) in virtually every aspect of their enterprise that has an impact on the end result.

Regarding future work, we aim to follow a multi-aspect approach for validating the main facets of the proposed research. We will pursue validation in diverse enterprise settings with different technical constraints and user requirements so that the impact is leveraged. Validation will be performed on a technical level (covering system-related metrics such as performance) and on a business level, covering the benefits for end-users of the proposed system. Specifically, on the business perspective, validation will be focused based on performance in terms of decreased maintenance costs and equipment deterioration and increased reliability. On the other hand, domain experts will validate the results based on factors which are usually hard to measure such as increase in safety, decrease of environmental impact and the added value of the proactive business intelligence capabilities provided. Validation of the approach will be performed in the context of the ongoing FP7 project ProaSense in two main use cases: proactive manufacturing in the area of production of lighting equipment, and proactive maintenance within the oil and gas sector.

Acknowledgments. This work is partly funded by the European Commission project FP7 STREP ProaSense "The Proactive Sensing Enterprise" (612329).

7 References

1. Hollnagel, E., Woods, D. D., & Leveson, N. (Eds.). (2007). Resilience engineering: Concepts and precepts. Ashgate Publishing, Ltd.
2. Dumbill, E. (2012). Planning for big data. O'Reilly Media, Inc.
3. Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. J Advanced Manuf. Technology, 50(1-4), 297-313.
4. Engel, Y., Etzion, O., & Feldman, Z. (2012). A basic model for proactive event-driven computing. 6th ACM Conf. on Distributed Event-Based Systems (pp. 107-118). ACM.
5. Fu, S., & Xu, C. Z. (2010). Quantifying event correlations for proactive failure management in networked computing systems. J. of parallel & distributed computing, 70(11), 1100-1109.
6. Huang H.-Y., Chou Y.-C. & Chang S. (2009). A dynamic system model for proactive control of dynamic events in full-load states of manufacturing chains. Journal of Production Research, 47(9), 2485–2506.
7. Georgiadis, P., & Michaloudis, C. (2012). Real-time production planning and control system for job-shop manufacturing: A system dynamics analysis. J. of OR, 216(1), 94-104.
8. Peng, Y., Dong, M., & Zuo, M. J. (2010). Current status of machine prognostics in condition-based maintenance: a review. J. Advanced Manuf. Technology, 50(1-4), 297-313.
9. Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mechanical systems and signal processing, 20(7), 1483-1510.
10. Bouvard, K., Artus, S., Berenguer, C., & Cocquempot, V. (2011). Condition-based dynamic maintenance operations planning & grouping. Application to commercial heavy vehicles. Reliability Engineering & System Safety, 96(6), 601-610.
11. Yao, X., Fernández-Gaucherand, E., Fu, M. C., & Marcus, S. I. (2004). Optimal preventive maintenance scheduling in semiconductor manufacturing. Semiconductor Manufacturing, IEEE Transactions on, 17(3), 345-356.

IT Governance in Organizations Facing Decentralization – Case Study in Higher Education

Jelena Zdravkovic[1], Irina Rychkova[2], Thomas Speckert[1]

[1] Department of Computer and Systems Sciences, Stockholm University, Forum 100, SE-16440, Kista, Sweden
jelenaz@dsv.su.se, thsp7525@dsv.su.se
[2] Centre de Recherche en Informatique, Université Paris 1 Panthéon - Sorbonne, 90 rue Tolbiac, 75013 Paris, France
irina.rychkova@univ-paris1.fr

Abstract. Decentralization of organizations and subsequent change of their management and operation styles requires changes in organization's processes and heavily involves the IT. A case study in the Higher Education sector in Sweden has shown that Enterprise Architecture (EA) frameworks fit to primarily centralized organizational structures, and as such have shortcomings when used in decentralized organizations. Overcoming these deficiencies requires some new principles to be introduced and incorporated into the EA knowledge. In particular for IT governance, the case study showed that the peer-to-peer principles, such as peer production, can offer more suitable governance over current EA frameworks as they are able to better match the decentralized components of the university's organizational structure.

Keywords: Enterprise Modeling, Enterprise Architecture, IT Governance.

1 Introduction

Enterprises have traditionally implemented formal, centralized forms of organizational structure [1], such as hierarchical or matrix structures. In these structures, communication patterns, roles and decision rights are strictly defined. This allows for management to have a high degree of control over the enterprise and therefore enforce compliance with standards, procedures and policies which results in a highly stable enterprise. However, this comes at the expense of agility; it is difficult for these organizations to quickly adapt to a changing environment. While centralized structures were appropriate for the business environments of the past, modern business environments demand a high level of agility.

Common components of modern business environments include cooperation with different organizations, rapidly changing business activities and processes, and a rapidly changing competitive landscape [2]. In order to properly handle these components, a high level of enterprise agility is necessary. In centralized organizations, decisions need to be discussed at all levels of the hierarchy in order to obtain the appropriate justification and approval. This takes time; by the time a decision is made, it is often too late for it to be effective. In contrast, having decision making on the operational level allows for quick decisions enabling an organization to take advantage of opportunities quickly. More decentralized structures, such as networked organizations [1], are examples of this. It is important to note that a lack of rigidity and formal structure does not mean a lack of organization. It is still important for a decentralized enterprise to maintain order in its activities; the governance (and IT governance) just needs to be based on an underlying decentralized structure instead of centralized one [3, 4].

Consequently, decentralized organizations need solutions to the same problems faced by centralized organizations – such as business-IT alignment – but the solutions need to be supportive of decentralization over centralization. This can be addressed by the practice of Enterprise Architecture (EA) [5].

Today's EA frameworks and methodologies need hence to be able to handle these environments, where rapidly changing business conditions have been identified as an important problem in EA in this context [6, 7]. For these reasons, ensuring the suitability of modern EA frameworks for decentralized organizational structures and governance which are highly dynamic, is becoming increasingly relevant.

This study reports the alignment between a decentralized organizational structure and an EA in use in a real organization; elicited problems are further analysis in respect to the support from current EA frameworks, as well as from other architectural principles that were considered to be able to solve the problems.

The paper is organized as follows: Section 2 reports a summary from a case study research in a Higher Education organization, in the requested STARR form: situation – task – approach – result – reflection. Section 3 provides conclusions and the directions of future work.

2 Case Study

The organizational structure defines the rules according to which allocation of responsibilities and resources, coordination and supervision, is made for an organization. Three key organizational properties differentiate between centralized and decentralized organizations: *geographical dispersion*, *coordination* (authority, decision rights, standards and regulations), and *communication patterns*. These properties were used as the base knowledge to assess the style of the case organization, and further to analyze the IT governance rules in place.

2.1 Situation

We have analyzed a prominent university for higher education in Sweden. As common, the university includes a number of units - faculties, and faculty departments. Nowadays, the units are becoming more independent than before, due to several factors:

- Geographical dislocation. Some faculty departments have been moved out of the main university campus. An example is the Computer and Systems Sciences department located in Kista, the leading Swedish IT cluster. This proximity enables cooperation between IT companies and students through mentoring programs, internships, graduate work opportunities, guest lectures, etc.
- Decentralization of management. Decision rights are of the type "push-down" delegated by the principal to the faculty boards and deans, and some to the faculty departments and their groups.
- Both formal and informal communication patterns. Formal hierarchical communication from the faculty to its departments and informal direct communication between and within the departments are present. For example, the administrative tasks (e.g. registration for graduate courses, or postgraduate research etc.) is primarily formal, whereas the course curriculum can be established between departments cooperatively, using informal communication links.

Hence, the organization is seen having high decentralization structure tendencies.

2.2 Task

The notions of *organizational structure*, *IT governance*, and *EA* are interrelated: EA principles should reflect the style of IT organizational structure; IT governance follows the organizational structure, and at the same time has to comply with the architecture to-be and the adopted EA principles.

EA includes governance processes such as IT principles regarding operations, data, architecture, infrastructure etc. They are to an extent similar to the processes of IT governance. However, EA governs the development and implementation throughout the organization directing the evolution of the IT and business environment towards a desired design of a future (i.e. primary strategic), while IT governance handles the everyday IT operations within the organization (i.e. primary operational).

The study was to analyze the aspects of university's EA in order to assess the decentralization support provided, in contrast with what is needed; to elicit conflicts between the architecture's principles in use, and the organizational structure and the governance rules, and thus provide a basis for the guidelines for an EA that can provide the needed support.

2.3 Approach

Four separate interviews were conducted in one of university's departments in order to get a holistic view of the way of work across the whole university. The roles of the interviewees were: vice division lead, head of postgraduate studies, head of undergraduate studies, and head of IT. The interviews were conducted in a semi-structured manner, starting with a set of open-ended questions that promote the interviewees to elaborate on their views to organization's processes, decision making, coordination, etc. In addition, many official documents are available on the organizational structure, thus making a document study viable. The documents that formed this study are described in Table 1:

Table 1. Documents used in the documentation study

Document	Description
Institution's homepage	Contains descriptions of the different organizational areas of the institution as well its organizational structure
Authority delegation documents	Publicly available documents specify authority and delegations of said authority of the insinuation's organizational units
Rule book	The official rule book of the institution detailing the rules and decisions that must be followed by the institution

2.4 Results

According to the EA related literature, enterprise architectural principles are established to define the general rules and guidelines for the use of assets across the enterprise. For the purpose of this study, we have chosen to concentrate on the following adopted EA principle:

- Integrated IT systems across the university.

Owing to a decentralized organizational structure described in 2.1 and as in more details uncovered during the interviews, some decision rights are pushed down to the operational level, which for the IT-related organizational structure has resulted in a highly *decentralized* governance:

Table 2. In-place IT governance framework

Name	Org. Property / Centralization	Description
Authority structure	Coordination / Decentralized	The department and the university have separate IT and the departmental IT does not report to the university
IT adoption (department)	Coordination / Decentralized	Department IT does not dictate all IT used in the department; research projects and centers; for example, groups can develop and use their own IT systems should they desire
Approval (department)	Coordination / Mixed	IT projects are run by independently by groups, though they sometimes need approval from the department if they are expensive

IT collaboration	Coordination / Decentralized	Any decision to cooperate with other departments or with the university IT is made by the departmental IT itself and is based on cooperation resulting on mutual benefit
Management of "essential" central IT systems	Coordination / Centralized	"Essential" systems (e.g. administrative systems such as HR) for the whole university are controlled by the university board. The department is required to pay for and use these systems.
Management of "non-essential" central IT systems	Coordination / Mixed	"Non-essential" systems (e.g. course portals and schedules) are centrally budgeted, but departments are not required to use them.
Use of IT systems (department)	Communication / Decentralized	Informal communication patterns are used, i.e. when changes are performed on systems, they are informally spread to those who use those systems.

In the practice, the governance structure described in the table has become in the mismatch with the settled EA principle to integrate IS systems. This mismatch has resulted in wasted financial resources. For example, we consider a situation outlined in the interview with the vice-head of the department which concerned the acquisition of a software system with the objective of integrated facility management across departments (i.e. "integrated systems" principle). Following the principle, a software system has been bought for university-wide use; since the principle holds for the whole enterprise, the purchase was the decision of the university-board, i.e. the departments were not involved in the decision making process. In contrast, following the decentralized IT governance in place for the use of "non-essential" software systems (Table 2), a subset of them consequently refused to shut down their local systems and switch to the global one. As a consequence, the principle of integration failed; the departments were able to protect their interests (local, decentralized systems tailored for their needs), but were still charged for the acquired system they never used.

Another important mismatch comes from the fact that the centralized decision making (i.e. faculty level) uses formal, hierarchical communication patterns, while decentralized, such as in case of IT governance relies on informal communication (see Table 2) which in practice has no supporting mechanisms. Hence, important decisions on changes in IT are not well communicated (not on time, or not at all) having severe working consequences for employees and students using it.

2.5 Reflections

The case has many of the properties of a decentralized organization and therefore needs an EA supportive of this decentralization. Currently this is not the case because:

- The EA is implicit and does not elaborate in details the adopted principles,
- The EA maintains some centralized principles and is therefore not fully supportive of the decentralization in place.

As a consequence, IT governance initiatives fail, and decisions in IT become inefficient.

Hence it has been relevant to investigate how existing EA frameworks are supportive for decentralized organizations. This question was systematically addressed in [8] where the three key organizational properties – a) geographical dispersion, b) coordination (authority, decision rights, standards and regulations), and c) communication patterns, were used to assess three wide-know frameworks - TOGAF [9], FEA [10], and Zachman Framework [11].

While the analysis revealed some support for decentralization, the main conclusion drawn is that the EA frameworks of TOGAF, Zachman, and FEA are primarily supportive of centralized (and federated) organizational structures, and therefore fail to address the demands of decentralized organizations. Zachman is unable to support any significant aspect of decentralization due to its reliance on traditional organizational roles and structures on the high centralization end. TOGAF does provide some basic support through its ability to have a different architecture for organizational units and by providing space for new methods for the architecture development; it however still mainly relies on hierarchy and central roles responsible for overall coordination and approval. In FEA, the conclusions are similar as it primarily supports federated organizational structures where individual units have their own architectures that are coordinated through centralized standards that must be followed. As shown earlier, an important property of a decentralized business environment that needs to be supported by EA is horizontal coordination (recall the IT governance from Table 2). However, the three EA frameworks primarily support vertical coordination in their governance styles. Therefore, the addition of specific guidelines to these frameworks that are supportive of decentralization would improve their support of decentralized organizations.

Drawing parallels between the domains of *peer-to-peer systems* used to provide a mechanism and architecture for organizing the peers in such a way so that they can cooperate to provide a useful service to the community of users [12] and decentralized organizations, we think that the peer-to-peer concept may be a source of principles that could form the basis for evolving current centralization-focused EA frameworks into ones that are supportive of decentralization.

- *Peer Production*: we view enterprises as being composed of peers (a peer could be individual or an organizational unit), For example, TOGAF relies on an Architecture Board responsible for high-level decisions and governance. Instead of a central board responsible for making decisions, a model based on the principle of peer production [13] for creation and evaluation of EA artifacts could be used instead. This would better support decentralization as decision making would then be distributed amongst the peers that make the organization. In the university case, the department members could produce strategy, or budget, using peer production (such as for use of information systems). Eventually, faculty or university boards could have control/advisory roles.

– *Peer trust management*: TOGAF employs the idea of an approval process grounded on the presence of centralized authority. This is to ensure that the presented architectural material is in fact valid for the enterprise. According to peer trust management [14], whether some content proposed by a peer is of sufficient quality to be included in the overall architecture, is determined by other peers. In the studied case, this principle could provide a formal mechanism for communication among peers when needed, hence avoid the situations when other peers are not informed about a new proposal (such as a change in IS use).

The suggested peer-to-peer principles will seek to maintain the departmental-independence becoming prevalent at the university, while addressing the incompatible architecture components this results in. This would be accomplished through a cooperative classification of essential and non-essential systems by the departments, for example by giving each department a vote. Systems classified as essential are required to be used or integrated by the departments, while departments have the option to choose if they want to utilize systems classified as non-essential. These changes would help at reconciling differences between the architecture principle emphasized in the case without actually changing it. Decision rights are still pushed down, and IT systems are still integrated throughout the organization, but this change in IT governance at the university level addresses the conflict that can arise when a decision is made to use a decentralized system that the rest of the organization is integrating (as occurred in the current situation).

3 Conclusion and Future Work

While technology serves as a catalyst for organizational transformations, it is important to utilize the right IT resources in a manner that is supportive for the organization. To accomplish this in decentralized organizations, adequate EA processes, principles and concepts are needed to be employed to both handle the IT resources and to foster business/IT co-evolution in decentralized environments.

Current EA frameworks rely on organizational properties that are becoming less useful with progressive decentralization. Due to this, implementation of these frameworks in decentralized organizations becomes difficult and inefficient, and the role of EA as a driver for IT transformations is becoming compromised. In order to deal with decentralization, some changes, or additions to these EA frameworks are necessary in order to improve their support for decentralized business environments, to reflect the style of organizational structure and operational IT governance rules in place Two specific principles of peer-to-peer architectures were outlined, peer production and peer-to-peer trust management; and indicated how they could be used as potential principles for an EA that is supportive of decentralization.

The reflections of this study may be of interest to three groups: the case organization, researchers in the field of EA, and, potentially, other organizations with decentralized structures interested in implementing some form of EA. For the case organization, the proposed EA principle of peer-to-peer might be of interest, as the application of this principle could offer some improvements to their governance structure. For researchers, this study work might be of interest as it highlights some potential issues with traditional EA knowledge, while giving some initial insights into how they could be solved. These insights are not conclusive; this research should be positioned as a starting point for future research in the topic of decentralization in EA. This work may be of interest to organizations that have adopted, or are interested in adopting a decentralized structure and are looking for the insights into how governance can be successfully done in this environment.

For the future work, we envisage to propose the concrete mechanisms and patterns for communication, coordination and decision making in centralized, decentralized and mixed (federated) organizations, and to see how they can be transformed into concrete EA principles, or explicitly integrated into EA methodologies.

References

1. Pearlson, K. E. and Saunders, C. S. *Strategic Management of Information Systems*, 4th ed. John Wiley & Sons, (2009)
2. Fulk, J. and DeSanctis, G. Electronic communication and changing organizational forms. Organization science, vol. 6, no. 4, pp. 337–349 (1995)
3. Rockart, J., Earl, M. and Ross, J. Eight imperatives for the new IT organization. Sloan management review, pp. 43–56, (1996)
4. Weill, P. and Ross, J. W. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*. Harvard Business School Press, Boston (2004)
5. Ross, J. W., Weill, P. and Robertson, D. *Enterprise Architecture As Strategy: Creating a Foundation for Business Execution*. Harvard Business Review Press (2006)
6. Lucke, C., Krell, S. and Lechner, U. Critical Issues in Enterprise Architecting - a Literature Review. in AMCIS 2010 Proceedings. (2010) At http://aisel.aisnet.org/amcis2010/305
7. Bente, S, Bombosch, U. and Langade, S. Collaborative Enterprise Architecture: Enriching EA with Lean, Agile, and Enterprise 2.0 Practices. Morgan Kaufmann, (2012)
8. Speckert T., Rychkova I., Zdravkovic J., and Nurcan, S. On the Changing Role of Enterprise Architecture in Decentralized Environments: State of the Art. To appear in proceedings of 8th International Workshop on Trends in Enterprise Architecture Research (TEAR). 9-13 September, Vancouver, BC, Canada (2013)
9. The Open Group, TOGAF Version 9.1. The Open Group, (2011)
10. Federal Enterprise Architecture Program Management. FEA Practice Guidance. (2007)
11. Zachman, J. A. John Zachman's Concise Definition of The Zachman Framework. (2008) At http://www.zachman.com/about-the-zachman-framework
12. Saroiu, S., Gummadi, P. K. and Gribble, S. D. Measurement study of peer-to-peer file sharing systems. In Electronic Imaging 2002. International Society for Optics and Photonics, pp. 156–170 (2001)
13. Benkler, Y. *The wealth of networks: How social production transforms markets and freedom*. Yale University Press (2006)
14. Aberer, K. and Despotovic, Z. Managing Trust in a Peer-2-Peer Information System. In Proceedings of the 10th International Conference on Information and Knowledge Management ACM, pp. 310–317 (2001)

Discovering Speech Acts in Online Discussions: A Tool-supported method

Itzel Morales-Ramirez[1,2] and Anna Perini[1]

[1] Software Engineering Research Unit. Fondazione Bruno Kessler - IRST
Via Sommarive, 18, 38100 Trento, Italy
imramirez,perini@fbk.eu
[2] University of Trento, Italy

Abstract. The increasing participation of users of software applications in online discussions is attracting the attention of researchers in requirements elicitation to look at this channel of communication as potential source of requirements knowledge. Taking the perspective of software engineers who analyse online discussions, the task of identifying bugs and new features by reading huge threads of e-mails can become effort demanding and error prone. Recognising discussants' speech acts in an automated manner is important to reveal intentions, such as suggesting, complaining, which can provide indicators for bug isolation and requirements. This paper presents a tool-supported method for identifying speech acts, which may provide hints to software engineers to speed up the analysis of online discussions. It builds on speech act theory and on an adaptation of the GATE framework, which implements computational linguistic techniques.

Keywords: Speech Acts, GATE framework, Intentions

1 Introduction

Researchers in requirements elicitation recognise that the increasing participation of users of software applications in online discussions turns these type of discussions into an attractive source of information. Such an information can be exploited for different purposes and one purpose can be to derive requirements knowledge. In this context we define requirements knowledge as the knowledge that contributes to the definition of software systems requirements, as well as to the modifications of requirements already specified.

In this line, if we take the perspective of software engineers who need to analyse probably many text files comprising the content of online discussions, the task of identifying bugs and new features by reading huge threads of messages or e-mails could make this task effort demanding and error prone. For instance, in an open-source software development that rests on distributed communities, open forums and mailing lists are communication means commonly used to enable the collaboration tasks to perform solution design, code writing, software deployment, maintenance and evolution.

Recognising discussants' speech acts in an automated manner can be seen as an important task to reveal intentions, such as suggesting, complaining, which are becoming of crucial importance to understand discussants' comments. We take the inspiration from the Speech Act Theory (SAT), originally formulated by Austin and Searle [1], and from the Grice's claim, in [2], that says "speaker's utterances automatically create expectations, which guide the hearer towards the speaker's meaning". Indeed, the speaker may aim at persuading, inspiring or getting the hearer to do something.

In this paper we present a tool-supported method that aids the discovery of speech acts, a task that we address using information extraction techniques. We believe that this identification of speech acts can provide software engineers hints to speed up the analysis of online discussions. Our approach builds on SAT and its application in computational linguistic [3]. Specifically, we exploited the GATE framework [4].

The remainder of the paper is structured as follows. In Section 2 we give some background on SAT and on the NLP framework used in our approach. In Section 3 we describe our tool that identifies speech acts as annotate intentions in text. The related work is presented in Section 4 and the conclusion in Section 5.

2 Speech Act Theory and GATE Framework

In this section we recall basic definitions from the philosophy of language, namely Speech Act Theory (SAT), we build on, and about the General Architecture for Text Engineering (GATE) tool for information extraction that has been adapted to implement our tool. The basic claim of the SAT developed by Austin and Searle in the field of philosophy of language [5, 2] is the following. When a person says something she/he attempts to communicate certain things to an addressee by getting him or her to be affected by the speaker's intention, in other words each utterance in a conversation corresponds to an action performed by the speaker. A speech acts involves then three types of acts, namely, locutionary, illocutionary and perlocutionary acts. A locutionary act is the act of "saying something", an illocutionary act makes reference to the way in which the locutions are used and in which sense, and a perlocutionary act is the effect on the audience that may be achieved. So, for instance, considering the utterance "I'll bring you a chocolate", the locutionary act corresponds to the utterance of this sentence, the illocutionary act corresponds to the speaker's intention to make the audience aware that she is committing to bring a chocolate, and the effect, i.e. the perlocutionary act, is that the audience got convinced about the speaker's intention.

Our tool makes use of a classification of speech acts that is proposed by Bach and Harnish in [1]. There are four main kinds of acts, namely constantives, directives, commissives, and acknowledgements. *Constantives* express the speaker's belief and her intention or desire that the hearer has or forms a like belief, e.g. "I must confess I'm a good chef". *Directives* express the speaker's attitude toward some prospective action that should be performed by the hearer and her intention that her utterance must be taken as a reason for the hearer's action. For example, if I say to you: "I want you to walk the dog in the evening", I intend to motivate you to perform the action. *Commissives* express the speaker's intention to commit to do something. As for instance when I say to you: "I am going to bring you a chocolate", I intend to make you believe that I'm committing to buy and bring a chocolate for you. Finally, *acknowledgements* express feelings regarding the hearer or the speaker's intention that her utterance satisfies a social expectation. For instance, "Please forgive me".

The GATE tool [4] is a framework, developed by the University of Sheffield in UK, for building and deploying software components to process human language. GATE can support a wide range of NLP tasks for Information Extraction (IE). IE refers to the extraction of relevant information from unstructured text, such as entities and relationships between them, thus providing facts to feed a knowledge base [6]. GATE is widely used both in research and application work in different fields (e.g. cancer research, web mining, law). This tool is composed of three main components for performing language processing tasks: *Language Resources* represent entities such as lexicons, corpora or ontologies; *Processing Resources* represent entities that are primarily algorithmic, such as parsers, generators or ngram modellers; and *Visual Resources* represent visualisation and editing components that are used in GUIs.

GATE offers the flexibility to replace or extend the *Processing Resources* component. For our purposes we have adapted the framework to build our tool by considering the following modules: (i) *Sentence splitter:* this module split the text into sentences, using RegEx splitter that is based on regular expressions; (ii) *Tokeniser:* is the module that identifies basic "tokens", such as words, punctuation symbols, and numbers; (iii) *Part-of-speech (POS) tagger:* this module associates tokens with parts of speech such as noun, verb, and adjective, based on the Hepple tagger [3]; (iv) *Morphological analyser:* this module is used to lemmatise the tokens and to provide words in their root form, e.g. running – run; (v) *Gazetteer:* this module can be adapted and it is a list of lists, where each list is a group of words that are associated with the domain; and (vi) *Java Annotation Patterns Engine (JAPE):* this is the main module that has been adapted in our tool and it enables the creation of rules in the form of regular expressions. The left-hand-side of the rule refers to what should match a fragment of text, and annotations in the right-hand-side says what should be done with the matched text.

3 Tool for Discovering Speech Acts

Our tool is based on a knowledge-heavy approach [7], this means the use of a POS tagger, JAPE rules, a tokeniser, a lemmatiser and gazetteers. In Figure 1 we present the different modules that have been used to build the tool. The bottom part of the figure shows Java as the platform on which GATE is built on and we have also reused to access to our dataset, XML is one of the input format that GATE allows and we have chosen to parse the processed files. On top of this layer are the modules that can be exchanged, and the JAPE and Gazetteers modules, which are flexible modules to be adapted according to the objective of the IE tasks. Gazetteers and JAPE rules have been tailored to annotate the intentions applying some tags that we have defined previously. These tags are used to annotate fragments of text, the tags are the subcategories of speech acts defined in Table 1. The first column with heading "Category" shows the main categories of speech acts and the column "Subcategory" is a specialisation used as the names for the tags to annotate. For instance, the first category *Constantives* is specialised into several subcategories, including the subcategory *Suggestives*, which corresponds to a linguistic act expressing the intention of a sender to make the receiver to consider as an option what he or she is suggesting. To adapt the JAPE module we

[3] Part-of-speech tags taken from http://gate.ac.uk/sale/tao/splitap7.html#x37-761000G

have formulated lexico-syntactic rules, by using a bag of words inspired from the given examples in [8] and by an empirical exploration of users' comments given in online platforms like the bugzilla issue tracking system.

The Gazetteers used in our approach are the lists of verbs taken mainly from [1] for each subcategory of speech act. Some JAPE rules use the Gazetteers to annotate intentions. The linguistic analysis is executed on discussion threads in the format of text files, which are the input. The tool is used to annotate intentions on the text messages of each thread. After this, the files annotated with intentions are parsed to extract the intentions found in each message. Finally, an analysis of intentions is performed, following an analysis model that we are elaborating.

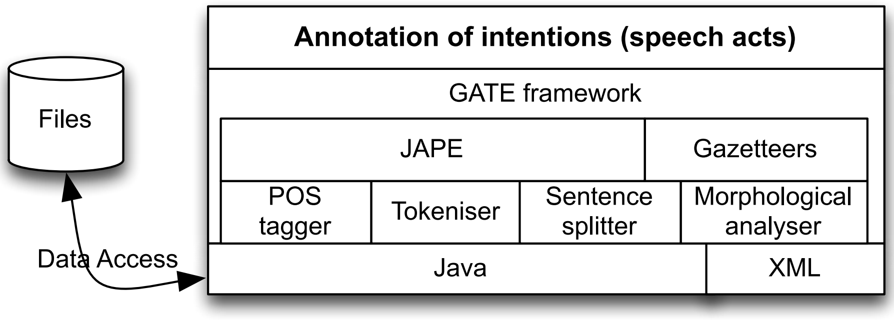


Fig. 1. Architecture of the tool.

Examples of design of JAPE rules are illustrated below, the full set of rules are available online[4]. The bottom part of the rule for tagging the speech act *Suggestives* represents the name of the tag to be annotated by the tool that processes the NL text, in our example the intention *Suggest*. In the second layer are the POS tags and tokens that are used by the tool to annotate such an intention. The POS tags $< PRP >$ and $< MD >$ refers to the initial set of words to annotate, the $< Keyword >$ refers to a list of verbs that we have defined in the Gazetteer modules and that are used by the JAPE rules[5].

$$\frac{\dfrac{\text{You}}{< PRP >} \quad \dfrac{\text{can}}{< MD >} \quad \dfrac{\text{“try”|“check”}}{< Keyword >}}{Rule\ to\ tag:\ Suggestives}$$

In the case of the rule to tag the speech act *Questions*, the second layer presents the starting and ending type of tokens that are used by the tool to annotate a question. In the left side of this second layer are the POS tags that can be found at the beginning

[4] JAPE files are available at http://selab.fbk.eu/imramirez/JAPErulesFeb2014/files.zip

[5] Gazetteer files are available at http://selab.fbk.eu/imramirez/GazetteerFeb2014/files.zip

Table 1. Categories of *speech acts* (excerpt of categories).

Category	**Subcategory**
Constantives	Assertives Concessives
	Suggestives
	Suppositives
	Responsives
Directives	Requestives
	Questions
	Requirements
Expressives	Thank
	Accept
	Reject
	Negative opinion
	Positive opinion
Attach (non-linguistic)	Link
	Code
	Log

of a question, i.e. ($< WRB >$ and $< VBZ >$), and in the right side, the *content* of a question plus the *question mark* indicating the end of a question. The top layer shows a concrete example.

$$\underbrace{\underbrace{\underbrace{\text{Where}}_{< WRB >}\quad \underbrace{\text{is}}_{< VBZ >}}_{Question\ begins}\quad \underbrace{\underbrace{\text{the option to delete}\ldots}_{< content >}\quad \underbrace{\text{“?”}}_{Question\ mark}}_{Question\ ends}}_{Rule\ to\ tag:\ Questions}$$

We manually designed the rules considering some characteristics for extracting the intentions, such as preceding and succeeding words, length of the words, root of the words, special types of verbs, using the bag of words, syntax and the codification of the POS tagger used by GATE. The tag is used to label a text fragment when one of the corresponding rule matches it. Each rule is formulated as a regular expression. The regular expressions $< content >$, $(< MD >)*$ and [Hh], for example, make reference to a set of words in the middle of two keywords or POS tags, to the presence or absence of the POS tag and to the uppercase or lowercase of the first letter of a word, respectively.

Each rule was then translated into JAPE rules, the tool uses the rules and tags to process the discussion threads in text format (i.e. txt files). This tool implements the modules of GATE as follows, see Figure 2 that depicts this process:

1. Cleaning and pre-processing: this process makes a document reset to clean the text of any previous tags, and noisy data.
2. Splitting into sentences: we use RegEx to split the text into sentences.
3. Splitting into tokens: we make use of a tokeniser provided in GATE to split the sentences into words.

4. Tagging tokens: then we run a POS tagger to tag each token into categories of nouns, verbs, punctuation, etc.
5. Lemmatising: the morphological analyser provided by GATE is used to lemmatise each word to its root, the example in the figure is the verb "wondering" into "wonder".
6. Gazetteers: in this process the Gazetteer' lists are used to identify specific verbs we have selected and that refer to intentions.
7. Applying rules: we use the JAPE rules to annotate the speech acts in the text. This is the last step that needs each word to be tagged with a POS tag, lemmatised and recognised by the Gazetteers.

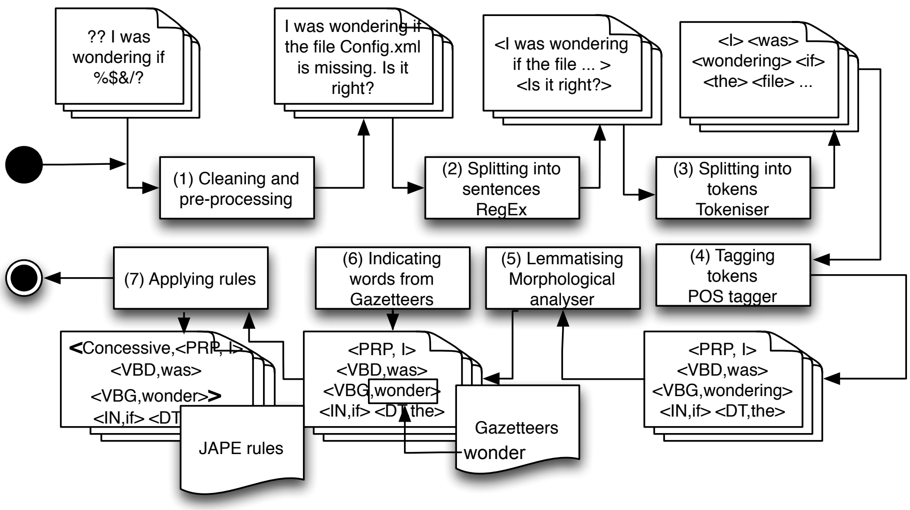


Fig. 2. Process to identify speech acts and annotate intentions.

After the annotation is executed, the tool parses the files to extract only the speech-acts tags and the corresponding intentions found in the text and a CSV file is generated by each discussion thread. Each file contains the discussants' name and the intention(s) identified in their messages.

We propose an analysis model of the intentions in a discussion thread that can be performed at different levels of granularity, namely, sentence and message level. At the sentence level we can identify single and nested speech acts. For instance, in the sentence "I suggest you to make a copy of your data", the single intention *Suggest* is the speech act "I suggest you", which refers to the subcategory *Suggestives*. An example of nested intentions is expressed in the sentence "Why don't you try to use the wizard?". In this case there are two speech acts, one is "Why don't you try to. . . ?" , and the other one is "don't you try", representative of the intentions *Quest* and *Suggest* respectively. At the message level the occurrences of pairs of intentions is analysed, called compound intentions, and we claim can be indicators of *Bug*, *Feature*, or *Clarification* requests. For example a combination of speech acts from the subcategory *Negative opinion* ("There is a problem") and *Question* ("Can anyone help me?") can be an indicator of a bug. Therefore, a set of nested or compound linguistic and non- linguistic acts can be considered as indicators of bug, features, and clarification. Currently, we are working on this model to incorporate it into the tool.

4 Related work

The analysis of NL textual messages in online discussion forums, bug-tracking systems or mailing lists has been addressed by research works that we briefly recall in this section. An automated identification of intentions is presented in [9]. This work proposes a tool that is based on SAT, dialogue acts and fuzzy logic to analyse transcripts of telephone conversations. One purpose of identifying intentions is that of detecting deception among participants in conversations by deriving participant profiles based on a map of the intentions expressed in such conversations. The classification of e-mails using speech acts is investigated in [10]. They are interested in classifying e-mails regarding office-related interactions as negotiation and delegation of tasks. Besides, they consider non-linguistic acts, such as *deliver*. In [11] the investigation of speech acts on thread of discussions, in student forum, aims at identifying unanswered questions to be assigned to an instructor for their resolution. They present some patterns of interaction found in the threads, the patterns correspond to the responsive and question speech acts.

With reference to Requirements Engineering tasks, Knauss et al. [12], analyse discussion threads for requirements elicitation purposes. They are focused on the content of communication between stakeholders to find patterns of communication used by stakeholders when they are seeking clarification on requirements. Their approach is based on a Naive Bayesian classifier, a classification scheme of clarification and some heuristics, with interesting results. Worth mentioning is also the work presented in [13] that aims at analysing messages, or *comments*, from users of software applications. Information extraction techniques and topic modelling are exploited to automatically extract topics, and to provide requirements engineers with a user feedback report, which will support them in identifying candidate new/changed requirements. All the above mentioned research works in the area of Requirements Engineering use NL text messages or documents to discover patterns, relevant topics or identify domain key terms, but none of the them consider SAT based techniques to understand stakeholders' intentions behind their messages. We consider that the application of SAT in Requirements Engineering can be a powerful strategy to understand stakeholder's intentions, thus supporting the analysis of the messages they exchange in current distributed collaboration and deriving requirements knowledge.

5 Conclusion

In this paper we presented a tool-supported method to identify speech acts in online discussions of OSS communities, which are developed by using unstructured natural language text. Indeed, we build on the idea that the recognition of discussants' speech acts is key to reveal their intentions, such as suggesting, or complaining. By automating speech acts recognition we may contribute to lower the software engineering effort in analysing huge discussions, also allowing for a better quality result.

The proposed approach exploits a taxonomy of speech acts based on the one proposed by Harnish and Bach. We illustrated the revisited taxonomy including linguistic and non-linguistic acts (such as code lines, URL links, and log files). Moreover, we described the adaptation of the framework GATE, specifically the modules Gazetteers and JAPE rules that can be configured according to the purpose of extraction, in our case the extraction of intentions. We explained the general process for annotating intentions and have explained the design of rules used to match fragment of text. Concerning the effectiveness of our approach, preliminary experimental evaluation on data sets extracted from online discussions in a OSS project are encouraging. We are currently planning further experimental evaluations to measure the accuracy of the tool by building a gold standard based on the annotations of three software developers. Our tool is currently based on a heavy-knowledge approach but in our future work we will target a machine learning approach.

References

1. Bach, K., Harnish, R.M.: Linguistic Communication and Speech Acts. MIT Press, Cambridge, MA (1979)
2. Wilson, D., Sperber, D.: Relevance theory. Handbook of pragmatics (2002)
3. Stolcke, A., Coccaro, N., Bates, R., Taylor, P., Van Ess-Dykema, C., Ries, K., Shriberg, E., Jurafsky, D., Martin, R., Meteer, M.: Dialogue act modeling for automatic tagging and recognition of conversational speech. Comput. Linguist. **26**(3) (September 2000) 339–373
4. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., Aswani, N., Roberts, I., Gorrell, G., Funk, A., Roberts, A., Damljanovic, D., Heitz, T., Greenwood, M.A., Saggion, H., Petrak, J., Li, Y., Peters, W.: Text Processing with GATE (Version 6). (2011)
5. Searle, J.R.: Speech acts: An essay in the philosophy of language. Volume 626. Cambridge university press (1969)
6. Cowie, J., Lehnert, W.: Information extraction. Commun. ACM **39**(1) (January 1996) 80–91
7. Ahrenberg, L., Andersson, M., Merkel, M.: A knowledge-lite approach to word alignment. In: Parallel Text Processing. Springer (2000) 97–116
8. Jurafsky, D., Shriberg, L., Biasca, D.: Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical report, University of Colorado at Boulder Technical Report 97-02 (1997)
9. Twitchell, D.P., Nunamaker, J.F., J.: Speech act profiling: a probabilistic method for analyzing persistent conversations and their participants. In: Proceedings System Sciences, 2004. (2004) 10 pp.
10. Carvalho, V.R., Cohen, W.W.: On the collective classification of email speech acts. In: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2005) 345–352
11. Ravi, S., Kim, J.: Profiling student interactions in threaded discussions with speech act classifiers. Frontiers in Artificial Intelligence and Applications **158** (2007) 357
12. Knauss, E., Damian, D., Poo-Caamano, G., Cleland-Huang, J.: Detecting and classifying patterns of requirements clarifications. In: RE 2012. (2012) 251–260
13. Carreño, L.V.G., Winbladh, K.: Analysis of user comments: an approach for software requirements evolution. In: ICSE, IEEE / ACM (2013) 582–591

Integrating Technical Debt into MDE

Fáber D. Giraldo[1,2], Sergio España[2], Manuel A. Pineda[1], William J. Giraldo[1] and Oscar Pastor[2]

[1] System and Computer Engineering, University of Quindío, Colombia
email: {fdgiraldo, mapineda, wjgiraldo}@uniquindio.edu.co
[2] PROS Research Centre, Universitat Politècnica de València, Spain
email: {fdgiraldo,sergio.espana,opastor}@pros.upv.es

Abstract. The main goal of this work is to evaluate the feasibility to calculate the technical debt (a traditional software quality approach) in a model-driven context through the same tools used by software developers at work. The *SonarQube* tool was used, so that the quality check was performed directly on projects created with Eclipse Modeling Framework (EMF) instead of traditionals source code projects. In this work, XML was used as the model specification language to verify in SonarQube due to the creation of EMF metamodels in XMI (XML Metadata Interchange) and that SonarQube offers a plugin to assess the XML language. After this, our work focused on the definition of model rules as an XSD schema (XML Schema Definition) and the integration between EMF-SonarQube in order that these metrics were directly validated by SonarQube; and subsequently, this tool determined the technical debt that the analyzed EMF models could contain.

Keywords: Model-driven engineering, technical debt, EMF, SonarQube.

1 Introduction

Two representative trends for the software development industry that appeared in the nineties were the *model-driven* initiative and the *technical debt* metaphor. Both trends promote software quality each in its own way: high abstract levels (models) and software process management (technical debt). However, despite the wide exposition of these trends in the literature, there are not more indications about the combination of them into software development scenarios; each initiative is implemented in a separated way.

In traditional software development projects (those involving manual programming), technical debt is mainly focused in quality assurance processes over source code and related services (e.g. common quality metrics are defined over source code). However, model-driven engineering (MDE) promotes for modelling instead of programming [1]. A review of the literature reveals that there is currently no application of the technical debt concept to environments outside the traditional software development. There exist approaches to the measurement of model quality [7][8][9][10], but these do not include technical debt calculus.

Therefore, we claim that dealing with technical debt in MDE projects is an open problem.

Two issues pose challenges to the inclusion of technical debt into MDE. *(i)* Different authors provide conflicting conceptions of quality in model management within MDE environments[5]. *(ii)* The MDE literature often neglects techniques for source code analysis and quality control[3]. Therefore, in model-driven developments it is difficult to perform an analysis of the state of the project that is important for technical debt management: establishing what has been done, what remains to be done, how much work has been left undone. Also, other specific issues that belong to model theory such as: number of elements in the metamodel, coverage for the views, complexity of the models, the relationship between the abstract syntax and the concrete syntax of a language, quantity of OCL verification code, among others, contribute to increase the technical debt in model-driven projects.

The main contributions of this paper are the following: *(i)* A demonstration of a integration between model-driven and technical debt tools for supporting a technical debt calculus process performed over conceptual models. *(ii)* The operationalization of a recognized framework for evaluating models.

2 Implementation of a technical debt plugin for EMF

We implemented an Eclipse plugin for integrating the EMF enviroment with SonarQube; so that, results of the technical debt can be shown directly on the Eclipse work area instead of changing the context and opening a browser with the SonarQube report. We used configuration options belonging to EMF *XMIResource* objects to export the XMI file as an XML without the specific XMI information tags.

2.1 Definition of an XSD for SonarQube

One of the most critical issues in a technical debt program is the definition of metrics or procedures for deducting technical debt calculations; in works like [4][6] it is highlighted the absence of technical debt values (established and accepted), and features such as the kinds of technical debt. Most of the technical debt calculation works are focused on software projects without an applied model-driven approach; some similar works report the use of high level artifacts as software architectures[12], but they are not model-driven oriented. Emerging frameworks for defining and managing technical debt [13] are appearing, but they focus on specific tasks of the software development (not all the process itself).

[3] Neglecting the code would seem sensible, since MDE advocates that the model _is_ the code [2]. However, few MDE tools provide full code generation and manual additions of code and tweakings are often necessary.

From one technical perspective, the SonarQube tool demands an XSD (XML Scheme Document) configuration file that contains the specific rules for validating the code; or in this case, a model. Without this file, the model could be evaluated like a source code by default. In order to define these rules, we chose one of the most popular proposals for validating models (*Physics of notations* of Moody [11]) due to its relative easiness to implement some of its postulates in terms of XSD sentences.

In the case of this work, visual notation was taken as the textual information managed by XMI entities from EMF models (text are perceptual elements too), focusing that each item meets syntactic rules to display each information field regardless about what is recorded as a result of the EMF model validation. The analysis does not consider the semantic meaning of the model elements to be analyzed.

The operationalization of Moody principles over the XSD file posteriorly loaded in SonarQube was defined as follows:

– *Visual syntax - perceptual configuration*: in the XSD file, it is ensured that all elements and/or attributes of the modelled elements are defined according to the appropriate type (the consistence between the values of attributes and its associated type is validated).
– *Visual syntax - attention management*: a validation order of the elements is specified by the usage of order indicadors belonging to XML schemes.
– *Semiotic clarity - redundant symbology*: a node in the model can only be checked by an XSD element.
– *Semiotic clarity - overload symbology*: an XSD element type only validates a single model node type.
– *Semiotic clarity - excess symbolism*: a metric to validate that there are no blank items was implemented (for example, we could create several elements of *Person* type, but its data does not appear).
– *Semiotic clarity - symbology deficit*: a validation that indicates the presence of incomplete information was made (e.g., we could have the data of a *Person* but we don't have his/her name or identification number). For this rule, we made constraints with occurrence indicators to each attribute.
– *Perceptual discriminability*: in the XML model, nodes must be organized in a way that they can be differentiated, e.g., one *Project* element does not appear like a *Person* element. This is ensured by reviewing in the XSD that it does not contain elements exactly alike, and in the same order.
– *Semantic transparency*: this was done by putting restrictions on the names of the tags, so that the tags correspond to what they must have, e.g, a *data* label must be of *data* type.
– *Complexity management*: this was done by the *minOccurs* and *maxOccurs* occurrence indicators. With these indicators it is possible to define how many children one node can have.
– *Cognitive integration*: this was done using namespaces in the XSD file, so that it is possible to ensure the structure for the nodes independent from changes in the model design.

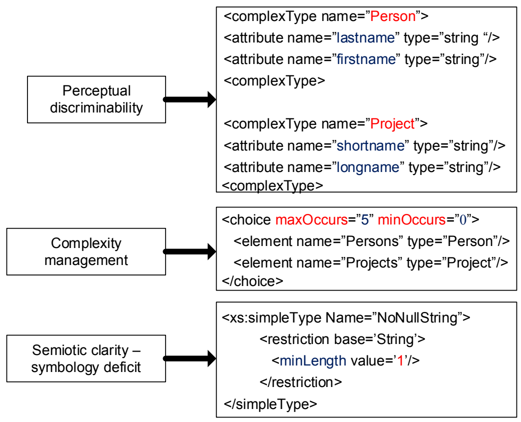


Fig. 1. Mapping between some Moody principles to XSD code.

- *Dual codification*: this was done by measuring the quantity of commented code lines with respect to the XML lines that define the elements of the model.
- *Graphic economy*: we established a limit for different items that can be handled in the XSD, and reporting when different elements are found marking the mistake when these data types are not found in the schema.
- *Cognitive fit*: this was done by creating several XSD files where each one is responsible for reviewing a specific view model.

Figure 1 exposes a portion of the XSD code implemented for some Moody principles.

2.2 Verification of technical debt from EMF models

In order to demonstrate the integration of both tools (EMF-SonarQube), a sample metamodel (Figure 2) was made in EMF[4]. We introduce some errors like capital letters, blank spaces, and others, to evidence abnormalities not covered with model conceptual validation approaches like OCL.

Once the validation option had been chosen (by the SonarQube button or menu), we obtain a report similar to Figure 3. Part *A* indicates the number of lines of code that have been tested, comment lines, and duplicate lines, blocks or files. Also, part *B* of this figure reports the total of errors that contain the project (in this case the EMF model), as well as the technical debt graph (part *C*), which shows the percentage of technical debt, the cost of repair, and the

[4] this metamodel was extracted from a web tutorial about EMF; source: http://tomsondev.bestsolution.at/2009/06/06/galileo-emf-databinding-part-1/

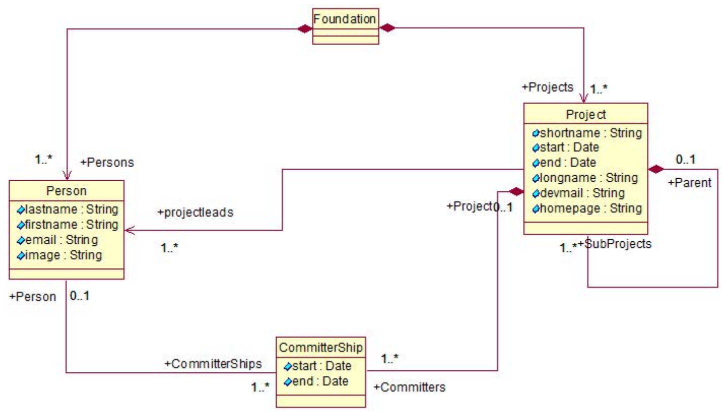


Fig. 2. Sample metamodel implemented over EMF.

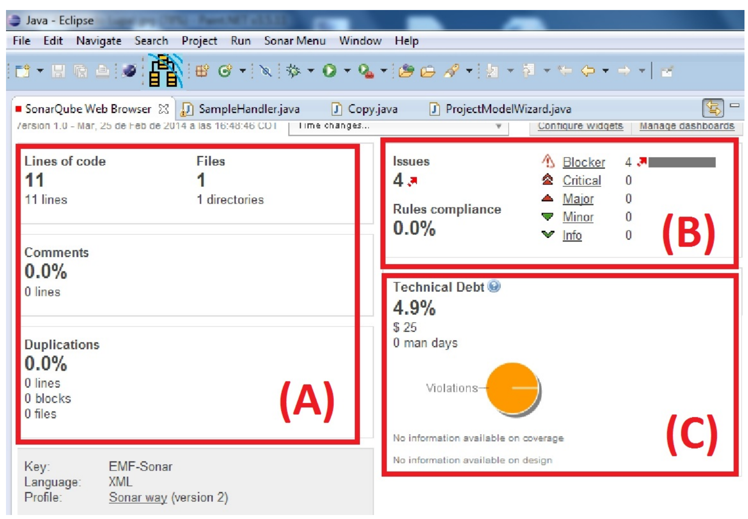


Fig. 3. SonarQube screen report loaded into EMF work area

number of men needed to fix errors per day (this information was not configured for this case).

SonarQube offers an *issues* report where it indicates the number of errors found; and consequently, the error list distributed in order of importance from highest to lowest:

– *Blocker*: they are the most serious errors; they should have the highest priority to review.

- *Critical*: they are design errors which affect quality or performance of the project (model errors can be classified in this category).
- *Major*: although these errors do not affect performance, they require to be fixed for quality concerns.
- *Minor*: they are minor errors that do not affect the operation of the project.
- *Info:* they are reporting errors, not dangerous.

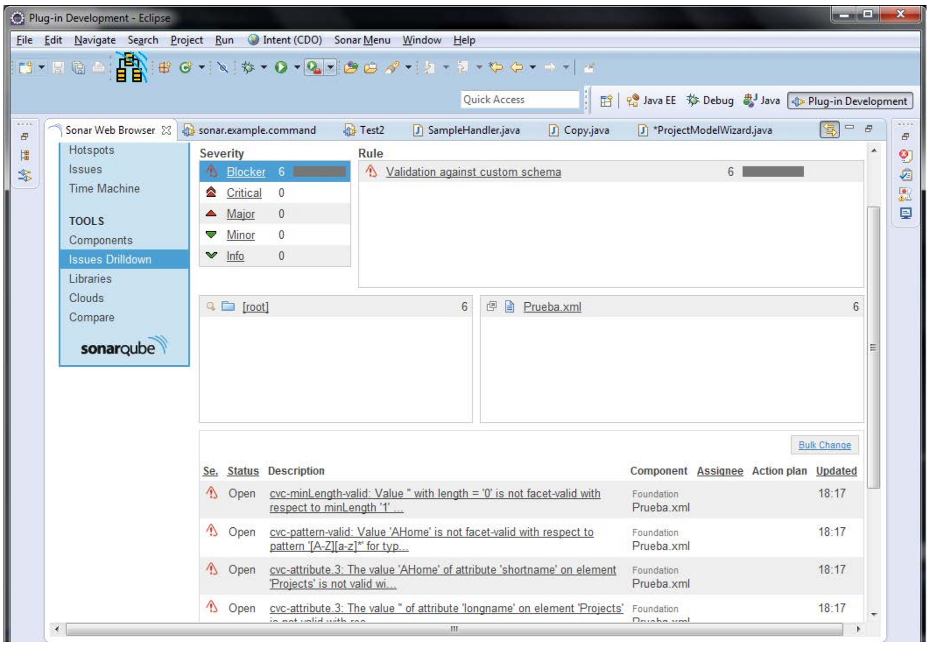


Fig. 4. SonarQube issues report of technical debt in EMF.

Figures 4 and 5 present the reports about technical debt errors detected over the sample model. In the first place, an error category was chosen. For the respective category, the error list associated is show in detail posteriorly. Intentionally, we introduced errors over the XML information of the model to test the respective detection by SonarQube according with the rules defined in the XSD file from the Moody proposal.

Conclusions

In this work we show the technical feasibility to integrate a technical debt tool like SonarQube with a model-driven development enviroment such as the Eclipse modelling framework. We present an example of technical debt validation applied

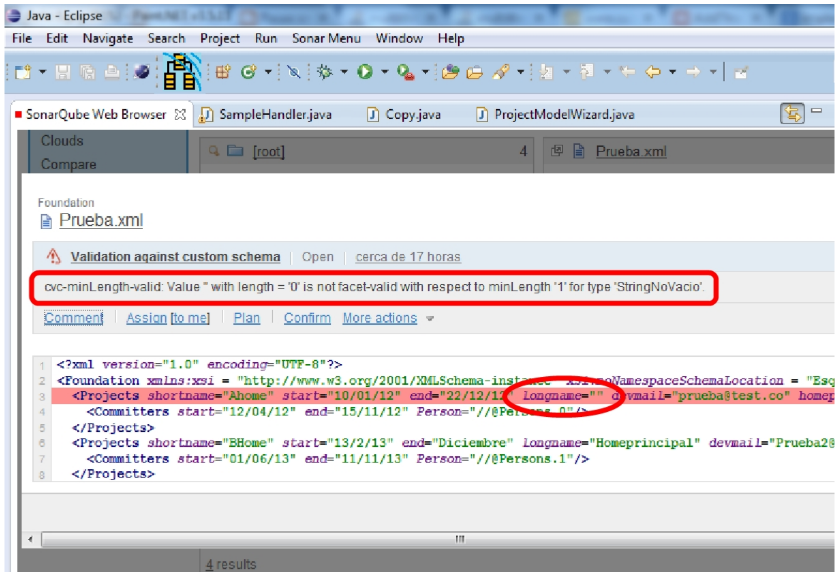


Fig. 5. Example of error detected by SonarQube.

over a sample metamodel implemented for testing purposes. Thereby, we demonstrate the technical feasibility for measuring any artefact used in an model-driven engineering process [3]. However, the main challenge is the definition of the model quality metrics and the operationalization of the model quality frameworks reported in terms of expressions that can generate metrics, and its association with a model-driven development process.

Acknowledgments

F.G, thanks to Colciencias (Colombia) for funding this work through the Colciencias Grant call 512-2010, and the Pan-American Software Quality Institute (PASQI) for the participation and feedback received in the PASQI Workshop 2013. F.G. and M.P. thanks to David Racodon (david.racodon@sonarsource.com) and Nicla Donno (nicla.donno@sonarsource.com) for their suppport with the SQALE plugin for SonarQube. This work has been supported by the Spanish MICINN PROS-Req (TIN2010-19130-C02-02), the Generalitat Valenciana Project ORCA (PROMETEO/2009/015), the European Commission FP7 Project CaaS (611351), and ERDF structural funds.

References

1. Marco Brambilla, Jordi Cabot, and Manuel Wimmer. *Model-Driven Software Engineering in Practice.* Synthesis Lectures on Software Engineering. Morgan &

Claypool Publishers, 2012.
2. David W. Embley, Stephen W. Liddle, and Oscar Pastor. Conceptual-model programming: A manifesto. In David W. Embley and Bernhard Thalheim, editors, *Handbook of Conceptual Modeling*, pages 3–16. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-15864-3.
3. Bertoa Manuel F. and Vallecillo Antonio. Quality attributes for software metamodels. In *Proc. of 13th TOOLS workshop on quantitatives approaches in object-oriented software engineering. QAAOSE 2010. July 2nd. Málaga, Spain*, February 2010.
4. D. Falessi, M.A. Shaw, F. Shull, K. Mullen, and M.S. Keymind. Practical considerations, challenges, and requirements of tool-support for managing technical debt. In *Managing Technical Debt (MTD), 2013 4th International Workshop on*, pages 16–19, 2013.
5. Peter Fettke, Constantin Houy, Armella-Lucia Vella, and Peter Loos. Towards the reconstruction and evaluation of conceptual model quality discourses - methodical framework and application in the context of model understandability. In Ilia Bider, Terry A. Halpin, John Krogstie, Selmin Nurcan, Erik Proper, Rainer Schmidt, Pnina Soffer, and Stanislaw Wrycza, editors, *BMMDS/EMMSAD*, volume 113 of *Lecture Notes in Business Information Processing*, pages 406–421. Springer, 2012. ISBN 978-3-642-31071-3.
6. C. Izurieta, I. Griffith, D. Reimanis, and R. Luhr. On the uncertainty of technical debt measurements. In *Information Science and Applications (ICISA), 2013 International Conference on*, pages 1–4, 2013.
7. John Krogstie. Quality of models. In *Model-Based Development and Evolution of Information Systems*, pages 205–247. Springer London, 2012. ISBN 978-1-4471-2935-6.
8. C.F.J. Lange and M.R.V. Chaudron. Managing Model Quality in UML-Based Software Development. In *Software Technology and Engineering Practice, 2005. 13th IEEE International Workshop on*, pages 7–16, 2005. LCCN 0029.
9. Beatriz Marín, Giovanni Giachetti, Oscar Pastor, and Alain Abran. A quality model for conceptual models of mdd environments. *Adv. Soft. Eng.*, 2010:1:1–1:17, January 2010.
10. Parastoo Mohagheghi and Vegard Dehlen. Developing a quality framework for model-driven engineering. In Holger Giese, editor, *Models in Software Engineering*, volume 5002 of *Lecture Notes in Computer Science*, pages 275–286. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-69069-6.
11. Daniel L. Moody. The ´physics´ of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, 2009.
12. R.L. Nord, I. Ozkaya, P. Kruchten, and M. Gonzalez-Rojas. In search of a metric for managing architectural technical debt. In *Software Architecture (WICSA) and European Conference on Software Architecture (ECSA), 2012 Joint Working IEEE/IFIP Conference on*, pages 91–100, 2012.
13. Carolyn Seaman and Yuepu Guo. Chapter 2 - measuring and monitoring technical debt. volume 82 of *Advances in Computers*, pages 25 – 46. Elsevier, 2011.

A Behavior Centered Modeling Tool Based on ADOxx

Judith Michael, Fadi Al Machot, Heinrich C. Mayr

Application Engineering Research Group, Alpen-Adria-Universität Klagenfurt, Austria
{judith.michael, fadi.almachot, heinrich.mayr}@aau.at

Abstract. Meta-modeling platforms that support the automatic generation of modeling tools open a new quality in information systems development for engineers: Emphasis can be put on the design and use of a modeling language that is customized to the particular needs and desired features. This may contribute to strengthen the information system design phase as it helps to reduce the developers' aversion against overloaded modeling languages and inflexible or expensive modeling tools. Our demo paper introduces HCM-L Modeler, a modeling tool for the Human Cognitive Modeling Language, which has been implemented using the meta-modeling platform ADOxx as a component of an ambient assistance information system.

Keywords: Conceptual Modeling, Individual Information System, Component based development, Prototype, Modeling Platform, Knowledge Management System, Meta-Modeling, Adaptive and Context-Aware System

1 Introduction

When thinking of Information Systems (IS), most images in mind are related to the business domain: providing support for managers and their decisions, supporting business processes and thus assisting the employees in their job functions. Typically, though mostly realized on the basis of an integrated standard system, IS are customized to the needs of respective enterprise and its users

Ambient assistance information systems for individuals, however, require an even more personalized functionality, which leads the notion of *self-centered IS*: A system for one particular person, assisting she/him by providing information about and from herself/himself, in a way tailored to her/his abilities and needs. At a first glance, this might sound irrelevant. But think about getting older and forgetting how to use a technical device, how to use the online banking software or even how to dress yourself on or how to cook your favorite dish: then you might wish to have individual assistance for mastering your activities of daily life in order to be independent from others. This leads us to the domain of Ambient Assisted Living (AAL) [1].

The AAL-project HBMS[1] aims at saving relevant information about human behavior of a person in a cognitive model (HCM, Human Cognitive Model) and providing

[1] The work is part of the HBMS project - Human Behavior Monitoring and Support: funded by Klaus Tschira Stiftung GmbH, Heidelberg - a research project in the field of Ambient Assisted Living

this information to the person when needed. To describe a person's individual HCM, the Human Cognitive Modeling Language HCM-L, i.e. a Domain-Specific Modeling Language (DSML), has been defined in order (a) to provide a user and use centered language and (b) to enable a mostly automatic model creation and integration out of sensor and/or tracking data. User centeredness should allow and simplify model validation and refinement when desired. As HCM-L is to describe behavioral ("episodic") knowledge, it can be called a conceptual cognitive modeling language.

This paper concentrates on the modeling tool supporting HCM-L, which forms a component of our HBMS-System (an ambient assistance IS) together with reasoning modules and a web-based support tool. As the tool primarily served as a proof-of-concept for the modeling language, the novelty of the approach lies mainly in that language. The future HBMS-system users of will be care givers and the supported persons themselves.

Section 2 briefly introduces the HCM-L using an example. Section 3 illustrates some features of the HCM-L Modeler. Section 4 outlines related work. Section 5 gives a resume and outlines future developments.

2 A Modeling Language for Ambient Assistance

We introduce the HCM-L only shortly here; more detailed information may be found in [2]. The HCM-L concepts were derived from analyzing the target AAL domain of (instrumental) activities of daily life [3] and their context [4]; the graphical notation considers the nine principles for designing cognitively effective visual notations [5].

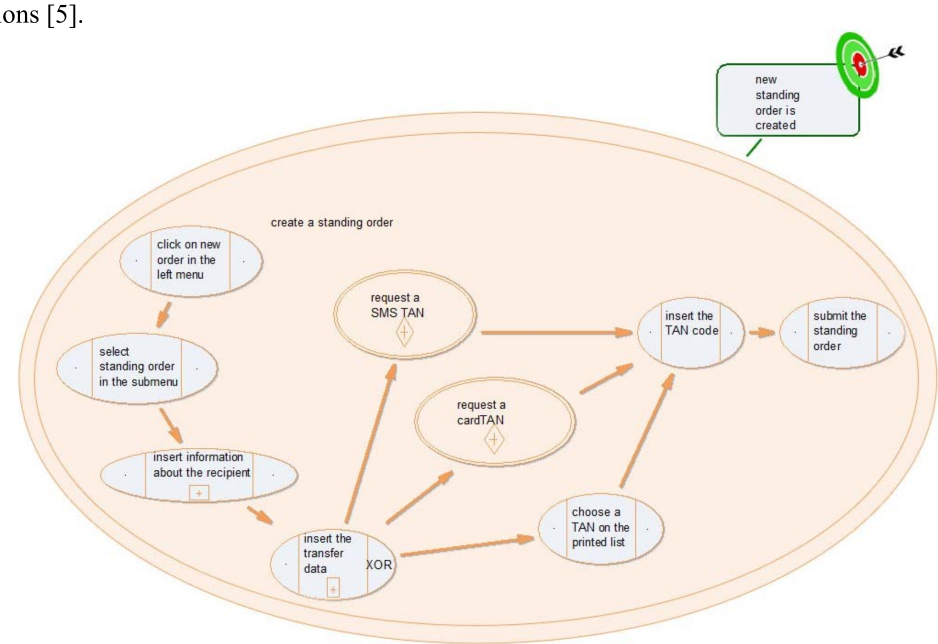


Fig. 1. Example BU for creating a standing order with several Operations and a Goal

Creating a HCM-L model starts from the most prominent elements in human behavior: activities. We call the resp. concept *Behavioral Unit (BU)*. Fig. 1 shows a BU 'create a standing order'.

Daily life activities usually have a goal which is reached by performing a sequence of actions. These actions are captured by the HCM-L concept *Operation,* graphically *drawn* inside the resp. BU (expressing that a BU 'consists' of operations) and linked by *Flows*. Having executed an operation without outgoing flow means that the BU's *goal* is reached, i.e. in our example: 'new standing order is created'. There may be alternative actions like the three ways to receive a Transaction Authentication Number (TAN); *Pre-* and *Post-Condition Expressions* allow arbitrary granularity for the control flow (graphically simply by naming the logical operator, see XOR in Fig.1).

Create a standing order may be part of a larger BU 'use the online banking system'; as well, more detailed information about actions may be needed for support; e.g. to request a SMS TAN, again a sequence of actions might be necessary. Therefore operations can be BUs, too. Thus, HCM-L allows for hierarchical structures.

Clearly, support information can not only be derived from dynamic structures. HCM-L, therefore, provides concepts for modeling structural contexts as well based one the areas described in [4]. For further information, please refer to [2].

A comprehensive control pattern-based analysis [6] revealed, that all relevant semantics can be expressed using HCM-L when modeling activities of human behavior, their hierarchies, and the relevant context information.

3 The HCM-L Modeler

The HCM-L Modeler was developed using the meta-modeling platform ADOxx®[2] [7] [8]. A main reason for choosing ADOxx was, that all basic modeling functions (drawing, linking and reorganizing elements, resizing, hierarchical arrangement, editing) could be implemented easily using the ADOxx Development Toolkit [7]. The HCM-L Meta-Model (also called user specific meta-model in the ADOxx context) inherits from the ADOxx Meta-Model. With ADOxx it was possible to define and realize the graphical notation, the different context models (see [2]) and further attributes of the elements of our modeling method in a notebook-representation.

In what follows, we outline some further features that go beyond these basic ones: model stepping for an animated walk-through, querying, checking the consistency of a model, providing reasoning support, reading sensor data for complex scenarios, as well as media file management.

Model Stepper

The stepper animates the succession of operations (of the active model) and allows a stepwise pass through a behavioral unit path based on users' decisions. Basically, this is achieved by highlighting the visited operation.

Once the stepper encounters the need of a user decision (evoked by a pre- or post-condition of the current operation), a selection window is opened where the user can

[2] www.ADOxx.org

choose the next step. In case of encountering a sub-unit (within a hierarchy of behavioral units) the stepper offers the choice between continuing on the current hierarchy level and walking through the sub-unit.

By visualizing the operation flows that are possible due to the model's structure, the stepper supports model understanding and validation. The long term idea is to provide this stepper functionality to end users in order to make validation possible for them.

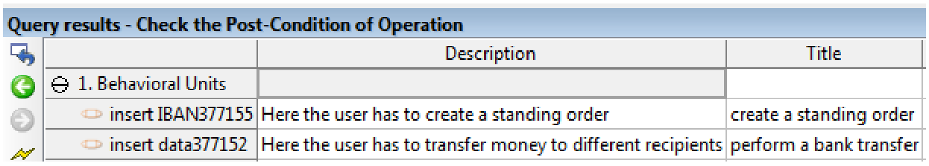
Querying and Predefined Queries

Based on the ADOxx querying feature HCM-L Modeler supports model validation based on (predefined) queries that are formulated using the SQL-like language AQL. Such queries may concern checking the values of attributes, the coherence of elements, the compliance with predefined rules and restrictions as well as the timing of events. AQL queries can be ad hoc formulated by a user or pre-defined by the metamodel developer in the Development Toolkit, e.g., a pre-defined query for event detection (information from sensor data). For ad-hoc formulation the HCM-L Modeler provides an interactive assistant using an ADOxx basic functionality.

As an example, the following query unveils all BUs in the given model that should occur between 06:00am and 11:30am. The user can create the following domain specific AQL query using the AQL queries window of HCM-L Modeler:

```
(<"Behavioral Unit">[?"atTime" >= "00:000:06:00:00"]) AND
(<"Behavioral Unit">[?"atTime" <= "00:000:11:30:00"]
```

Although this query is not terribly realistic in our running example, we show the HCM-L Modeler result in Fig. 2 in order to give an impression on how the system operates: the tabular output consists of the IDs, descriptions and the titles of all behavioral units that should occur between 06:00 and 11:30.



Query results - Check the Post-Condition of Operation

	Description	Title
1. Behavioral Units		
insert IBAN377155	Here the user has to create a standing order	create a standing order
insert data377152	Here the user has to transfer money to different recipients	perform a bank transfer

Fig. 2. Answer to the AQL (possible behavioral units between 06:00 and 11:30)

Clearly, queries may be more complex by addressing value type restrictions for attributes or complex events in the sense of aggregations of simpler or atomic ones.

Consistency Check

A major issue in modeling processes is the fact that comprehensive consistency checks are difficult, in particular for inexperienced users. However, inconspicuous mistakes in the logic may affect the whole model: contradictory semantics reduce the performance of reasoning processes and yield invalid results. For the HCM-L Modeler we considered three main consistency issues: (1) using the right syntax of logical operators, (2) consistent naming of model elements throughout the whole model and

(3) the overall syntax check during modeling to allow the right connection between different types of classes and relation classes. Whereas (2) and (3) are automatically checked during the modeling process, (1) is accomplished using the AQL feature: After clicking on the button "pre-defined queries", HCM-L Modeler yields a menu of different consistency checks for every model and sub model, e.g., checking the correct syntax of the pre-condition lable of an operation or operation-makro. Further consistency check-queries are in preparation.

Figure 3 shows the result of the consistency check "*Post-Condition of Operation with Pre Post and Suboperations*". Apparently there were problems with the post conditions of two operations ("check the TAN number" and "Insert data").

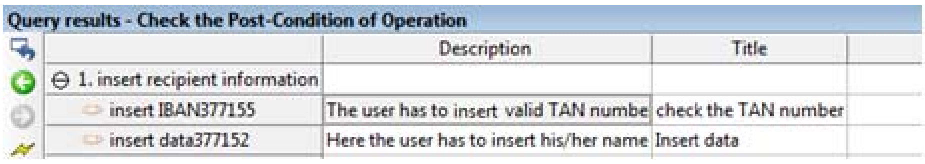


Fig. 3. Result of the condition check

Reasoning Support

Both model and rule based reasoning approaches for behavior modeling requires the extraction of different features out of the given overall model. HCM-L Modeler, among others, offers the possibility to calculate the frequency of specific activities based on the user history: every operation is supported by a percentage value.

In addition to that, the HCM-L Modeler calculates for each operation the "importance value" based on the user history, and (if wanted by a user) the "cost value" based on the similarity between the current user profile and other users.

Furthermore, it delivers for every operation the smallest number of the remaining operations (i.e., operations to be executed) until reaching the current BU's goal, together with all possible paths leading to that goal and under consideration of all subunits (e.g., see Fig. 4). I.e., the possible next steps are shown (for loops only one loop-iteration). As the selection of the next step is always based on the user's decision, loops are no problem.

As ADOxx offers the possibility to import and export models in a generic XML format, all those reasoning attributes can be used, e.g. by external inference or reasoning tools.

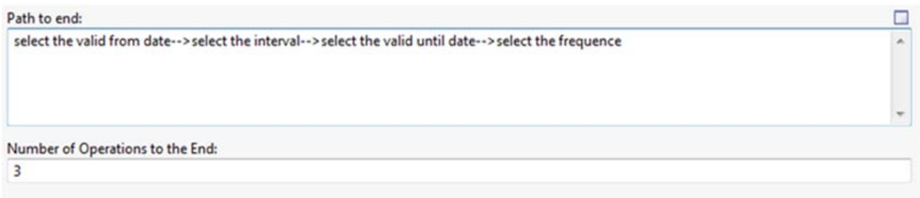


Fig. 4. Path to the end (see fig.1) from the current operation "select valid from date"

Reading Sensor Data

As already mentioned, the HCM-L Modeler is a part of a HBMS-System with different components. User monitoring will be provided through run-time by using sensors. This sensor data will be used to create the models using HCM-L (firstly simple sequences and after integration more generalized models).

ADOxx provides means to read content from files and databases to be included in the model (object or model level). It can read text, CSV, XLS, XML and DB formats. The HCM-L Modeler currently uses this feature for importing sensor data that are provided in XML.

The file should contain the ID of the sensor, the state of the action (true or false) and the time stamp of the selected activity. For user convenience we included predefined AQL queries into the HCM-L Modeler to simply check active operations and their states.

Media Files

The HCM-L Modeler offers the possibility to upload media files (video, audio and images files in different formats) into the tool. This feature allows using such files for visualizing complex issues and situations in the support phase (web-based support tool of the HBMS-System).

For example, if the user has to insert the card security code (CSC), sometimes called card verification data (CVD), the corresponding picture is presented to the user (automatically or after request) to show where this code is printed on the card.

Experiences Using ADOxx

Generally, developing a modeling tool by use of a meta-modelling platform proved to be a good way for implementing a tool with basic functionalities in a short period of time. In particular, ADOxx turned out to be an appropriate platform for DSML modeling tool development.

Moreover, the support provided by the ADOxx experts was helpful to implement the desired functionalities. They provided helpful examples additionally to the ADOxx standard tutorial.

Despite of the previous advantages we still have more complex requirements that are not implemented yet because of limitations of the meta-meta model definition, e.g., a visualization of static and dynamic elements in one view with a stepping functionality or the generation of predefined model element instances for a certain scenario (instances of living room, dining room and kitchen for the AAL domain).

4 Related work

Several projects concern about activity recognition in the AAL domain. For example, [9] use smart meters to detect activities of daily living; [10] show how behavior tracking can help to address different cognitive deficits based on plan recognition; [11] introduces an ambient intelligent living assistance system for mapping of real time sensor data to activities of a person.

Regarding modeling approaches, most related research and development endeavors also are in favor of using DSMLs (see, e.g., [12]).

As AAL is a rapidly growing domain, many projects aim at providing support for people, e.g. [13] for remembering the past, remembering to perform an intended action (e.g. take a medication), or to do cognitive training; [14] uses Case Based Reasoning techniques for solving support cases in a similar way to recently performed ones; [15] focus on modeling of personal goals and user characteristics to identify a possible impact on the system goals in general. However, none of these approaches supports a comprehensive recognition and exploitation of a person's basic and instrumental daily behavior.

5 Outlook

As already been mentioned, the HCM-L Modeler is just one of the HBMS information system components. The main access point for the end users will be the support component. For this, the models are transformed into a step-by-step description which can be displayed on an appropriate device. A first prototype was implemented and tested with 40 people in 2012 [16]. A beta version of the support component is currently under development and evaluation.

The support component uses the Operations' label, description and media files (like video, audio and images) to display a single behavioral step. In one of our end-user studies we investigated the best way of presenting information to them by using a set of mock-ups. The results showed that a combination of pictures and audio information is the preferred presentation form [17].

As we have shown, the HCM-L Modeler is a powerful and comprehensive tool for developing, managing and exchanging models written in HCM-L. The next development steps for the HCM-L Modeler will focus on the design of advanced reasoning approaches, model optimization, model checker, complex event detection and sensor data fusion with respect to sensors' uncertainty.

Furthermore, we will work on the model visualization layers to show the overall model architecture (all models and sub-models with respect to the structural context) in 3D to give the modeler and the software developers the possibility to understand the models and the interaction between them more easy.

The support component will be further tested, and we will pay more intention on individual users' preferences. [15] provides here some interesting ideas. Other improvements will concentrate on the definition of the support texts following previous work in computational linguistics [18]. The idea of automatic support text generation from a model (see [19]) seems to be an interesting approach.

References

1. Steg, H. et al.: Europe Is Facing a Demographic Challenge. Ambient Assisted Living Offers Solutions. VDI/VDE/IT, Berlin (2006)

2. Michael, J., Mayr, H.C.: Conceptual Modeling for Ambient Assistance. In: Ng, W., Storey, V.C., Trujillo, J. (eds.) Conceptual Modeling - ER 2013, LNCS 8217, pp 403-413, Springer (2013)
3. Katz, S.: Assessing self-maintenance. Activities of daily living, mobility, and instrumental activities of daily living American Geriatrics Society, New York NY (1983)
4. Kofod-Petersen, A., Mikalsen, M.: Context: Representation and Reasoning, Special issue of the Revue d'Intelligence Artificielle on "Applying Context-Management" (2005)
5. Moody, D.: The “Physics” of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Trans. Software Eng. 35, 756–779 (2009)
6. Mayr, H.C., Michael, J.: Control pattern based analysis of HCM-L, a language for cognitive modeling. In: International Conference on Advances in ICT for Emerging Regions (ICTer2012), pp 169-175, IEEE (2012)
7. Fill, H.G., Karagiannis, D.: On the Conceptualization of Modelling Methods Using the ADOxx Meta Modelling Platform. Enterprise Modelling and Information Systems Architectures 8(1), pp 4-25 (2013)
8. ADOxx® Software Engineer Tutorial Package. URL: http://www.adoxx.org/live/tutorial; [last visited on September 29, 2013]
9. Clement, J., Plönnigs, J., Kabitzsch, K.: Intelligente Zähler: ADLs erkennen und individualisieren. Smart Meter: Detect and Individualize ADLs. In: Technik für ein selbstbestimmtes Leben. Tagungsbeiträge. VDE-Verlag, (2012)
10. Giroux, S., Bauchet, J., Pigot, H., Lussier-Desrochers, D., Lachappelle, Y.: Pervasive behavior tracking for cognitive assistance. In: Proc. 1st ACM Int.Conf. on PErvasive Technologies Related to Assistive Environments - PETRA '08. ACM Press (2008)
11. Nehmer, J., Becker, M., Karshmer, A., Lamm, R.: Living assistance systems. an ambient intelligence approach. In: Proc. 28th Int. Conf. on Software Engineering, pp. 43-50. ACM, 1134293 (2006)
12. Izquierdo, J.L.C., Cabot, J.: Enabling the Collaborative Definition of DSMLs. In: Achour-Salinesi, C.B., Norrie, M.C., Pastor, O. (eds.) Advanced information systems engineering. CAiSE 2013 proceedings, LNCS 7908, pp. 272-287, Springer (2013)
13. González, M.F., Facal, D., Navarro, A.B., Geven, A., Prost, S., Tscheligi, M.: External memory support for cognitive care, results from the first Hermes user trial. In: Ipiña González de Artaza, D.L., Bravo, J. (eds.) Proc. Workshop of Ambient Assisted Living IWAAL, pp. 17-24. Garceta, Madrid (2010)
14. Zhou, F., Jiao, J.R., Chen, S., Zhang, D.: A Case-Driven Ambient Intelligence System for Elderly in-Home Assistance Applications. IEEE Trans. Syst., Man, Cybern. C 41, 179-189 (2011)
15. Sutcliffe, A., Sawyer, P.: Modeling Personalized Adaptive Systems. In: Achour-Salinesi, C.B., Norrie, M.C., Pastor, O. (eds.) Advanced information systems engineering. Proc. CAiSE 2013, LNCS 7908, pp. 178-192. Springer, (2013)
16. Gratzer, W.: Ontologiegestützte Assistenz durch mobile Endgeräte im Kontext von AAL, Master Thesis. Universitätsbibliothek Klagenfurt, (2013)
17. Strobl, T., Katzian, A.: Darstellungsvarianten für handlungsunterstützende Apps: Ergebnisse einer Evaluationsstudie im Rahmen des Projekts HBMS, Wohnen-Pflege-Teilhabe - 7. Deutscher AAL-Kongress, Berlin (2014)
18. Fliedl G., Gratzer W., Strobl T., Winkler Ch.: Mobile Instruction Apps Based on Linguistic Text Reduction. In: Rault, J.-C. (ed.) Proc. ICSSEA 2013, Paris (2013)
19. Leopold, H., Mendling, J., Polyvyanyy, A.: Generating Natural Language Texts from Business Process Models. In: Ralyté, J. et al. (eds.) Advanced Information Systems Engineering. Proc. CAiSE 2012, LNCS 7328, pp. 64-79. Springer, New York (2012)

UPROM Tool: A Unified Business Process Modeling Tool for Generating Software Life Cycle Artifacts

Banu Aysolmaz[1] and Onur Demirörs[2]

[1, 2]Informatics Institute, Middle East Technical University, Ankara, Turkey
[1]banu@aysolmaz.com, [2]demirors@metu.edu.tr

Abstract. UPROM tool is a business process modeling tool designed to conduct business process and requirements analysis in a unified way to constitute a basis for process automation. Enabling the application of UPROM methodology, the tool provides editors for six different diagram types based on a common meta-model. It ensures conformance to the rules and offers features so that modelers can develop a cohesive set of models. These models are utilized to automatically generate artifacts of requirements document, software size estimation, process definition document and business glossary.

Keywords: Business process modeling, requirements analysis, software size estimation, business glossary, process documentation, artifact generation

1 Introduction

UPROM tool is a business process modeling (BPM) tool supporting six diagram types: Value Chain (VC), Function Tree (FT), extended Event Driven Process Chain (EPC), Organization Chart (OC), Conceptual Entity Relationship (ER) and Function Allocation (FA). Diagrams represent different business process perspectives which are functional (VC and FT), behavioral (EPC), organizational (OC) and data (ER and FA).

Business process definitions serve both pure organizational purposes like process improvement and other practices like project management, requirements specification and knowledge management [1]. Usually in the organizations, artifacts that utilize business process knowledge are developed independently. Eventually, the effort spent to define business processes and develop related artifacts is duplicated, organizational knowledge becomes untraceable to processes and artifacts are maintained separately resulting in conflicts. Especially when business processes are to be automated by a process aware information system (PAIS) [2], process knowledge is intensively needed. To overcome the problems, we developed a unified BPM methodology, UPROM, to conduct business process and requirements analysis in an integrated way. When UPROM is applied, following artifacts can be generated by using the models: user requirements document and COSMIC functional size estimation [3] for the PAIS, and process documentation including process definition document and business glossary.

UPROM tool is a graphical BPM tool that supports UPROM methodology and automatically generates the mentioned artifacts. Model driven approach is followed based on Eclipse Modeling Framework (EMF) and Eclipse Graphical Modeling Framework (GMF). Eclipse plugins are developed for editors. All editors are based on a common meta-model. Some plugins including EPC and VC editors were reused from bflow* Toolbox [4], thus inheriting its special features such as continuous verification.

UPROM tool is used by process modelers for descriptive analysis of processes in business domain. End users utilize it to review and validate the models. The tool provides specific functionality for modelers to integrate business process and requirements analysis. There are tools that can generate process documentation, but we did not encounter a BPM tool generating textual requirements and functional size estimation. UPROM tool was utilized in various projects, including two e-government projects for Company and Trademark Central Registration Systems, Public Investment Analysis of Ministry of Development, and other applications for case study purposes.

In this paper, we present the features particular to UPROM tool to support the methodology and generate the artifacts. In section 2, UPROM methodology is briefly presented. Section 3 describes the tool features and provides a brief comparison with other tools. Section 3 summarizes the paper and presents the future work.

2 UPROM Methodology

UPROM methodology aims to integrate the practices of descriptive business process definition, requirements analysis, software size estimation and process documentation; and generate artifacts that are outputs of these practices. By unifying analysis activities for these practices, a set of models can be developed that embeds all information to generate those artifacts. The methodology includes the notation, meta-model, process, guidelines and artifact generation procedures. The artifacts that can be generated by UPROM methodology are: user requirements document, software functional size estimation, process definition document and business glossary. As all of these artifacts are based on a single source of model set, completeness and consistency of them are improved, they become traceable to business processes and maintainability is enhanced. More information on UPROM methodology and outputs of case studies can be found in [5] and [6]. UPROM methodology is applied in two iterative phases:

Developing Core BPM Diagrams: Functional, behavioral, and organizational perspectives of business processes are analyzed. As a result, VC, FT, EPC and OC diagrams are developed in a hierarchical manner.

Developing Analysis Diagrams Associated to BPM Diagrams: If a function on an EPC diagram is to be automated by PAIS, an FA diagram is created as a sub-diagram. FA diagram is used to analyze the responsibilities to conduct the function, related entities, operations on entities, related applications and constraints. In parallel, conceptual definitions of entities and their relations are modeled in ER diagram.

Generated artifacts are utilized as inputs to subsequent phases of software development. User requirements with process models and documentation are inputs to detailed requirements analysis, testing and acceptance phases. Functional size estimation is critical for software development planning in early phases. Process definition document and business glossary are used by different stakeholders types in operation phase.

3 UPROM Tool

UPROM tool provides an integrated environment for the notation and common meta-model for six diagram types. Diagrams of the same scope are maintained under a repository as a "modeling project". Features particular to UPROM tool enable users to apply UPROM methodology process and guidelines. Such features are described in the following sections. A snapshot of the modeling environment can be seen in Fig. 5.

3.1 Diagram Editors for BPM and Requirements Analysis

UPROM tool editors run in conformance with the meta-model. VC diagram comprises value chain, risk, objective and product constructs. FT diagram has only the function. EPC include event, function, process interface, logical operators, business rule, application, organizational elements, information carriers, key performance indicator (KPI), technical term and improvement [7]. OC covers organizational elements. FA diagram has organizational elements, function, entity, application and constraint. ER diagram covers entity, cluster, attribute, generalization, aggregation and relationship.

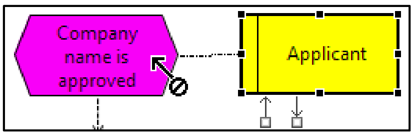


Fig. 1. Disabled connections based on the metamodel

Relations between constructs are restricted by the meta-model. The tool prevents formation of a connection not allowed and informs the user with a sign as shown in Fig. 1. Predefined connection names are assigned between some constructs. Two examples are shown in Fig. 2. First one is the connection assigned between an organizational element and function to show the responsibility of the role, and the second is between function and entity to specify operations conducted on the entity by the function.

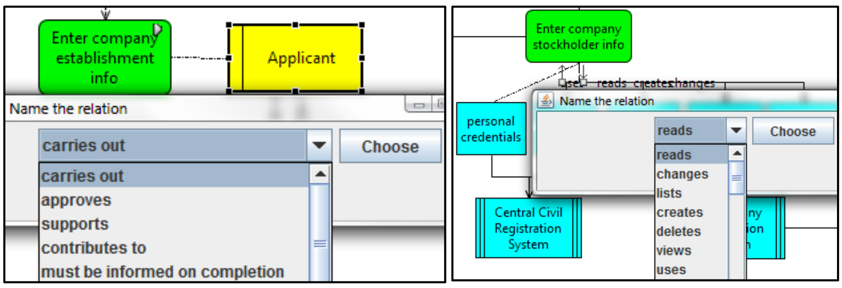


Fig. 2. Assigning connection names by selection from the combo box

3.2 Unique Object Assignment

Objects in the same modeling project are assigned to be unique if they are named the same. Instances of the object can exist at any diagram regardless of the type. Objects of logically similar types are also assigned to be unique; e.g. information carrier and entity; function, process interface and value chain objects with the same name are unique.

When a new object is added, if there is already an object with the same name and type (or one of the alternative types) in the project, the user is asked if the new object is the same with existing object(s) as shown on left of Fig. 3. If the user approves that they are the same, attribute values of the objects are assigned to be the same. When attributes of any instance of the unique object are updated, all other instances also have the updated values. Users can search for the occurrences of unique objects, see the list of instances and open the diagrams as shown in the right part of Fig. 3.

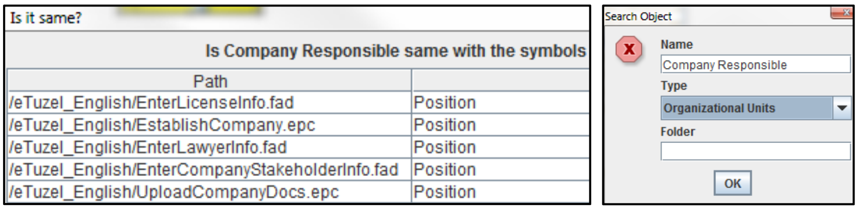


Fig. 3. Assigning unique objects with a list of instances and searching for the instances

3.3 Process and Object Attributes

A set of attributes representing metadata of a process can be assigned to each diagram as shown on left in Fig. 4. All objects have the attributes of name, id, description, incoming and outgoing connections. In addition to these, some object types have special attributes. One is technical term attribute for organizational elements, information carriers, application, entity, cluster, KPI and attribute. The other is document link attribute for information carriers, entity, cluster, business rule and constraint. Sub-diagrams can be assigned to function, process interface and value chain objects. An example list of attributes assigned to a document object is shown at right of Fig. 4.

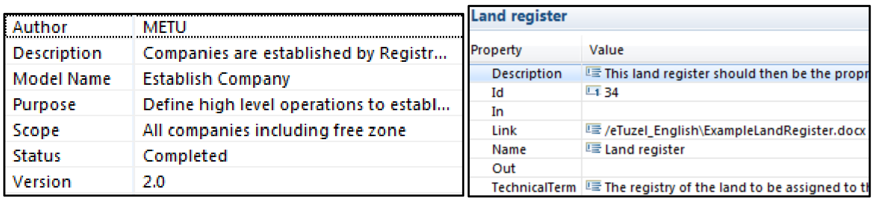


Fig. 4. Process metadata and object attributes

3.4 Structure of the Modeling Project

Folder structure of the modeling project must be established in conformance to its sub-diagram decomposition. The tool controls the folder structure and does not generate artifacts if it is not validated. An example modeling project structure is shown on the left part of Fig. 5 (FA diagrams are hidden for simplicity). Only one diagram of type VC, FT or EPC exists in top level, which is the process map (e.g. eCompany.ftd). For each sub-diagram assigned in the process map, a folder is created and the sub-diagram file is placed inside it. Folder and file names match. The same rules apply for lower level diagrams. FA is assigned as sub-diagram for the functions in EPCs, and placed under the folder of the related EPC. ERD and OC diagrams can be placed anywhere.

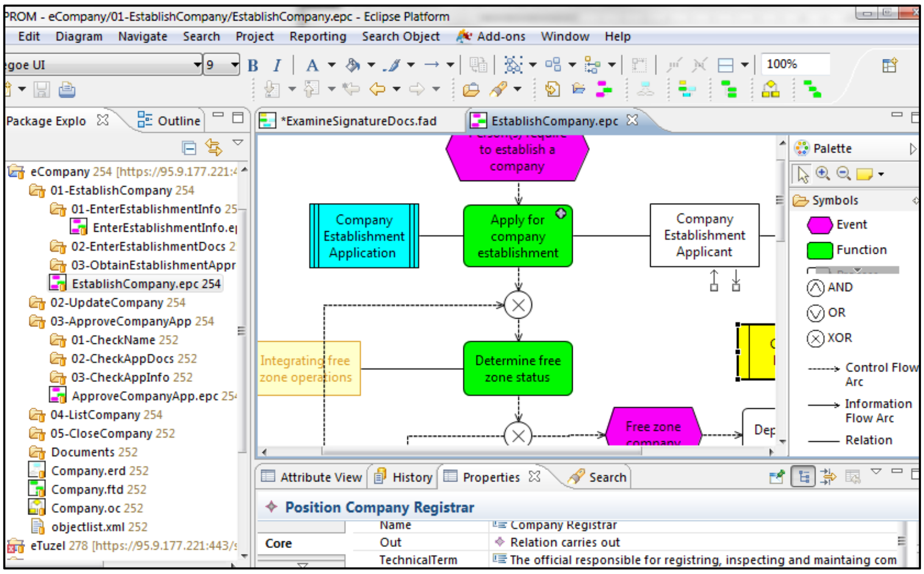


Fig. 5. Typical UPROM modeling environment and example project structure

3.5 Generation of Artifacts

Utilizing the information embedded in the modeling project which is formed by using the features and conforming to the rules explained in the above sections, UPROM tool can be used to generate the following artifacts. The tool parses XML diagram files and generates the artifacts in PDF format using iText library.

Business process models report: VC, FT and EPC diagrams in the modeling project organized by the hierarchy are reported. Name and address of the process are placed as the heading for each diagram. OC diagram(s) are placed at the end of the report.
Analysis models report: FA diagrams are given in this report. For each diagram, model name, address, and the EPC process it is related to are placed as the heading.
Requirements document: Each FA diagram is utilized to generate three types of natural language requirements sentences [6]. In the document, requirements sentences are organized under EPC diagrams. Headings for each EPC diagram is numbered according to diagram's hierarchical position in the modeling project.
COSMIC functional size estimation report: FA diagrams which serve the purpose of requirements analysis are also utilized to make an early functional size estimation of the software to be automated. The estimation is based on COSMIC standard [3]. Various rules are applied to interpret the operations on entities (modeled as connection name

between function and entity as shown in Fig. 2). For each EPC diagram, data movements and total size in function points (FP) of every FA diagram under that EPC is reported. The total FP size of each application is calculated and provided in the summary section of the report.

1.1. Process name: EstablishCompany.epc
Process address: eCompany/01-EstablishProcess
1.1.1. REQ1. Company Establishment Applicant shall cary out the operation of applying for company establishment.
1.1.2. REQ2. During this operation, application status and company records shall be created on Company Establishment Application System.
1.1.3. REQ3. During applying for company establishment, company status shall be assigned as "data entrance".
1.1.4. REQ4. Company Establishment Applicant shall cary out the operation of selecting company type.
1.1.5. REQ5. During this operation, company type record shall be created and updated on Company Establishment Application System.
1.1.6. REQ6. During selecting company type, company type shall be selected from a list of types determined in the legislation.
1.1.7. REQ7. Company Establishment Applicant shall cary out the operation of determining free zone status.

1.1. Pocess name: EstablishCom
Process Address: eCompany/01-
1.1.1. MSR1. ApplyforCompanyEstab
Company Establishment Application:
Entry Function Point: 1
Read Function Point: 0
Write Function Point: 2
Exit Function Point: 1
Total: 4 CFP
1.1.2. MSR2. SelectCompanyType
Company Establishment Application:
Entry Function Point: 2
Read Function Point: 1
Write Function Point: 2
Exit Function Point: 1
Total: 6 CFP

Fig. 6. Excerpts from Requirements Document (left) and Size Estimation Report (right)

- **Process definition document:** VC, FT and EPC models, with the attributes embedded in the models, are utilized to generate process definition document conforming to a template. The parts of the document for an EPC model is as follows:
  - Information: Purpose, scope, status, version, author, description of the process
  - Responsibilities: Name, type and participation of the role
  - Inputs: Name, type, source (if input is provided by another role), document link
  - Outputs: Name, type, target (if outputs are handed to another role), document link
  - Entrance criteria: Name, other processes that exit with this event, their address
  - Exit criteria: Name, other processes that start with this event, their address
  - Activities: Detailed information including: responsibilities, inputs and outputs, application, sub-diagrams/external processes, detailed information
  - Business rules: Name, related activity
  - External processes and sub-processes utilized by the process
  - KPIs: related activity, information sources used, measurement period, target
- **Business glossary**: All definitions in the project are obtained from the models by using technical term attributes of organizational element, information carrier, application, entity and cluster objects. By means of unique object property, an object has single definition regardless of the number of instances. The report is composed of three parts: Organizational, application and general definitions. ER diagram is utilized to organize general definitions. An aggregate entity is placed at the top, left indented. The components of the aggregate entity are grouped under that and indented right. Relationships and generalizations are also shown in a similar manner.

2. Establish Company

/01-EstablishCompany/EstablishCompany.epc/

1 Process Information

Purpose: Define the high level process to establish a company
Scope: All companies to be registered as general and free zone company
Status: Approved by customer
Version: 1.0

2 Responsibilities

		Responsibility Type				
Responsible	Type	R	A	S	C	I
Company Establishment Applicant	External Person	X				
Company Registrar	Position	X				

3 Inputs

Name	Type	Source	Link
declaration of registry	Document	Company Establishment Applicant	Documents\Ek4-6.ms1.pdf
clearance record	Document	Company Establishment Applicant	

4 Outputs

Name	Type	Target	Link
Approval of documents	Log		

5 Entrance Criteria

Event	Processes that exit with this event	Address
Person(s) require to establish a company		

6 Exit Criteria

Event	Processes that start with this event	Address
Applicant is informed for the inappropriate documentation		
Company establishment completed	Obtain Establishment Approvals	

7 Activities

7.1 Apply for Company Establishment

Responsibles: Company Establishment Applicant - carries out

Application System: Company Establishment Application System
Any citizen registering as a user of e-Company system with her ID info can apply for the company establishment. To establish a private company two or more people shall apply.

2 Application Systems

Name	Definition
Company Establishment Application	The system on which application for a new company establishment is conducted, updated and status updates viewed.

3 General Definitions

Name	Definition
Company	Any company established and registered as 1949 Company Law.
company communication info	Address of main and branch office, web address, phone, GSM, e-mail
company fundamental info	all fundamental info regarding the company
business domains	domains selected from NACE codes.
application no	
company name	
free zone status	

Fig. 7. Excerpts from Process Definition Document (up) and Business Glossary (down)

Excerpts from the generated artifacts are shown in Fig. 6 and Fig. 7. Complete versions of the outputs for multiple case studies can be seen in [5]. UPROM tool is developed specifically to apply UPROM methodology. Before developing the tool, we tailored diagrams in ARIS Business Architect [7] to meet meta-model needs and developed scripts to generate the artifacts. However, this did not provide a native solution to apply UPROM methodology. As an alternative to Eclipse, we could also use a modeling language creation tool such as MetaEdit+ [8] to design and use the diagrams based on the meta-model and generate the artifacts, however we preferred an Eclipse based system as we were able to reuse plugins from bflow Toolbox. There are tools that provide abundant notation alternatives such as ARIS. In the contrary, diagram types of UPROM tool are limited and focused on structured analysis of descriptive process models and requirements in an integrated way as guided by the methodology. There are other BPM tools with process documentation functionality such as Signavio, Bizagi and Visual Paradigm. However to our knowledge, there are no tools that can generate textual requirements and functional software size for the process automation together with the process documentation and providing guidance by a methodology.

4 Conclusion and Future Work

In this paper, we presented unified BPM tool. UPROM tool is based on EPC for control flow modeling, and supports five other diagram types. It provides an integrated modeling environment for requirements analysis in relation with business process models. If one applies UPROM process and guidelines by using tool features, she can use the tool to automatically generate some essential artifacts for software development practices.

UPROM is used in two e-Government projects of Company and Trademark Central Registration Systems and Public Investment Processes of Ministry of Development. Generated artifacts were used as project deliverables and acceptance is completed by the users [5]. The results collected by observations and interviews revealed that completeness, consistency and maintainability of the generated artifacts were improved.

For future versions, we plan to develop a functionality to enable users design the format and content of the artifacts and add new process documents presenting process information from different perspectives like RACI charts. At the moment, requirements sentences are generated in Turkish and generation of English sentences are planned for future versions. BPM tools supporting EPC notation are rather restricted in number. EPC is commonly accepted as a good notation for analyzing processes with end users. We believe that similar functionality can be achieved for also BPMN and plan to implement the methodology in a similar way also based on BPMN.

Acknowledgements. UPROM Tool was developed as a research project in Bilgi Group Ltd. New version with extended features is being developed with partial support of Industrial R&D Projects Support Program of the Scientific and Technological Research Council of Turkey with project no 3130770.

References

1. Becker, J., Rosemann, M., Uthmann, C. Von: Guidelines of Business Process Modeling. Business Process Management (2000). pp. 30–49. Springer (2000).
2. Dumas, M., Aalst, W. V d., Hofstede, A. ter: Process-aware information systems : bridging people and software through process technology. John Wiley & Sons, New Jersey (2005).
3. Abran, A., Desharnais, J.-M., Oligny, S., St-Pierre, D., Symons, C.: The COSMIC Functional Size Measurement Method Version 3.0.1 Measurement Manual. (2009).
4. Böhme, C., Hartmann, J., Kern, H., Kühne, S.: bflow* Toolbox-an Open-Source Modeling Tool. Business Process Management Demonstration Track 2010. pp. 1–6 (2010).
5. Aysolmaz, B.: Unified Business Process Modeling Methodology (UPROM) Application: Case Study Results METU/II-TR-2014-32. , Ankara, Turkey (2014). http://expertjudgment.com/publications/METU_II_TR_2014_32.pdf
6. Aysolmaz, B., Demirörs, O.: Modeling Business Processes to Generate Artifacts for Software Development : A Methodology. Modeling in Software Engineering (MISE), 2014 ICSE Workshop on. , Hydarabad, India (2014). DOI: 10.1145/2593770.2593775
7. Davis, R., Brabander, E.: ARIS Design Platform Getting Started with BPM. Springer London (2007).
8. Tolvanen, J., Pohjonen, R., Kelly, S.: Advanced Tooling for Domain-Specific Modeling : MetaEdit +. Object-Oriented programming systems, languages, and applications (OOPSLA) (2007).

Demonstration of the Online Method Engine

Kevin Vlaanderen, Sandor Spruit, Fabiano Dalpiaz, and Sjaak Brinkkemper

Utrecht University,
Department of Information and Computing Sciences,
P.O. Box 80.089,
3508 TB, Utrecht, the Netherlands,
{k.vlaanderen,a.g.l.spruit,f.dalpiaz,s.brinkkemper}@uu.nl

Abstract. During the past decade, much research has been performed in the areas of method engineering and process improvement. As a result of this research, we are developing the online method engine (OME). The OME is a knowledge management system that provides support during process improvement initiatives, using a set of assessments, an extensive method base, and automatic method assembly mechanisms. The system is designed around four main acivities: knowledge dissemination, method assessment, method improvement, and method enactment. In this paper, we demonstrate the core components and the main scenarios for the usage of the online method engine.

Keywords: method engineering; software tools; online method engine

1 Introduction

In an attempt to describe and reuse software development practices, several researchers have proposed method bases to capture this knowledge [5, 11]. The method fragments in these method bases are being configured and re-engineered using several techniques [8, 12], and implemented through a variety of tools [6]. Some of these techniques are in turn applied to the method engineering activity itself [4].

The online method engine (OME) is the implementation of scientific concepts in the field of method engineering (ME) and software process improvement (SPI), in combination with concepts from the field of knowledge kanagement (KM)). The main goal of the OME is to provide a system that facilitates the sharing of knowledge related to method fragments for software development, assessing methods that are being used in practice, improving methods based on assessment results, and adopting (improved) methods in practice. The main philosophy behind the OME is that incremental (or stepwise) improvement of processes reduces risks related to process change and improves the change of success [10, 3, 1].

Following the four goals of the OME, the system is based on four layers of functionality; knowledge dissemination, method assessment, method improvement, and method enactment. Each layer increasingly relies on the previous ones, and all of the layers rely on a central method base, which forms the backbone of the system. The quality of the system is ensured by a constant process of validation and review.

2 Architecture

Traditionally, method bases contain fragments of method knowledge, which reside on the M2 level (according to the MOF framework) [9]. In the context of the OME, the method base contains some additional elements. In the first place, it contains a set of situational indicators that can be used to describe an organizational unit. These situational indicators describe the specifics of the organizational context in terms that are unrelated to the specific method implementation. The situational indicators can be used for other purposes, but this will be described later on.

Complimentary to the situational indicators, the method base contains a set of capabilities that can be used to characterize a specific method implementation in a specific domain. The capabilities relate directly to a domain model, which captures the relevant process areas in a specific domain.

The situational indicators and the capabilities are related through a domain-dependent mechanism called the situational factor effect. A situational factor effect describes how a specific value of a situational indicator typically influences the relevance of a capability [2]. These situational factor effects are also captured in the method base.

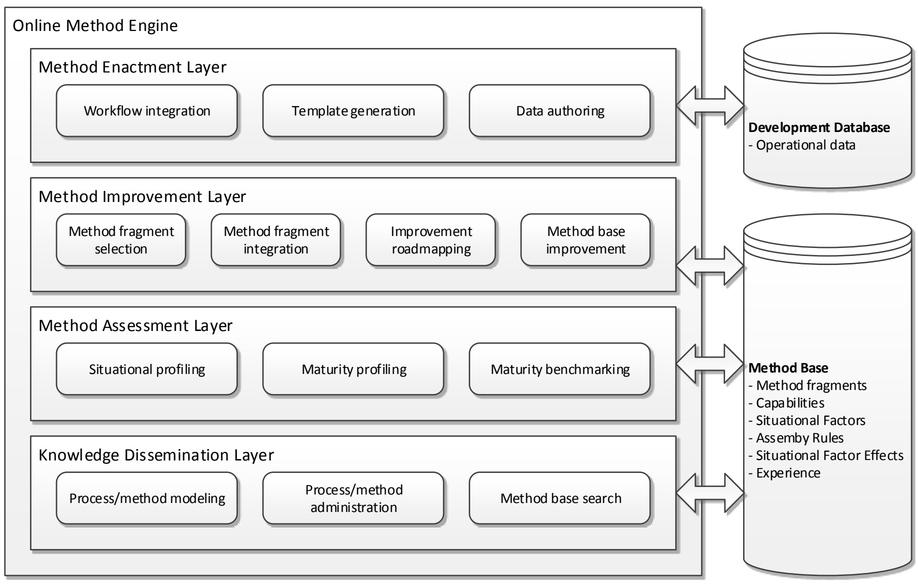


Fig. 1. Functional Architecture of the OME [17]

The final core element of the method base consists of a set of method fragments. Method fragments describe a coherent piece of method knowledge that facilitates reaching a specific goal during software development. A method fragment is described in terms of activities and deliverables. Within the OME, method fragments are currently modeled using process deliverable diagrams (PDDs) [18], which are a combination of a UML activity diagram and a UML class diagram, connected through a set of links.

3 Scenarios

3.1 Scenario A: Method Assessment

Maturity	A	B	C	D	E	F
Requirements management						
Requirements gathering	Implemented	Implemented	Missing	Missing	Missing	Irrelevant
Requirements identification	Implemented	Implemented	Missing	Irrelevant		
Requirements organizing	Implemented	Missing	Missing			
Release planning						
Requirements prioritization	Implemented	Implemented	Irrelevant	Implemented	Missing	
Release definition	Implemented	Missing	Missing	Irrelevant	Irrelevant	
Release definition validation	Missing	Missing	Irrelevant			
Scope change management	Implemented	Missing	Irrelevant	Irrelevant		
Build validation	Implemented	Implemented	Missing			
Launch preparation	Implemented	Implemented	Implemented	Irrelevant	Missing	Missing
Product planning						
Road map intelligence	Implemented	Implemented	Implemented	Missing	Missing	
Core asset roadmapping	Implemented	Implemented	Missing	Irrelevant		
Product roadmapping	Implemented	Implemented	Missing	Irrelevant	Missing	
Portfolio management						
Market analysis	Implemented	Implemented	Implemented	Implemented	Missing	
Partnering & contracting	Implemented	Implemented	Implemented	Missing	Irrelevant	
Product life cycle management	Implemented	Missing	Missing	Irrelevant	Irrelevant	

Fig. 2. Example Areas of Improvement Report

In general, a process improvement effort starts with an assessment of the current processes. The assessment approach employed within the OME is based on the situational assessment method (SAM) [2]. This method consists of three phases; data collection, calculcation, and feedback. Data collection is performed through two questionnaires; one to determine the organizational context, and one to determine the current capabilities of the organization. During the calculcation phase, the former results are transformed into a current capability profile (CCP) and the latter into an optimal capability profile (OCP). The delta between these two profiles results in an areas of improvement matrix (AIM). During the feedback phase, an evaluation is performed that is used to improve the quality of the knowledge base (i.e. the method base).

The SAM is realized through two forms within the OME. The first form is used to capture the organizational context. It consist of 24 questions spread out over 5 pages. Each question has a short description, a set of answers, and possibly some help text to indicate the type of answer expected. The second form is used to capture the current capabilities. This form contains 68 questions, which are spread out over 4 pages. Each page represents a business function, i.e. a layer from the SPM competence model [19]. The results of the questionnaires can be reviewed using two seperate reports; a situational profile report and a capability maturity report. The combined results are shown through an areas of improvement report, which shows both a condensed as well as an expanded version of the AIM. An example of the condensed *areas of improvement* report is shown in Figure 2.

3.2 Scenario B: Method Discovery

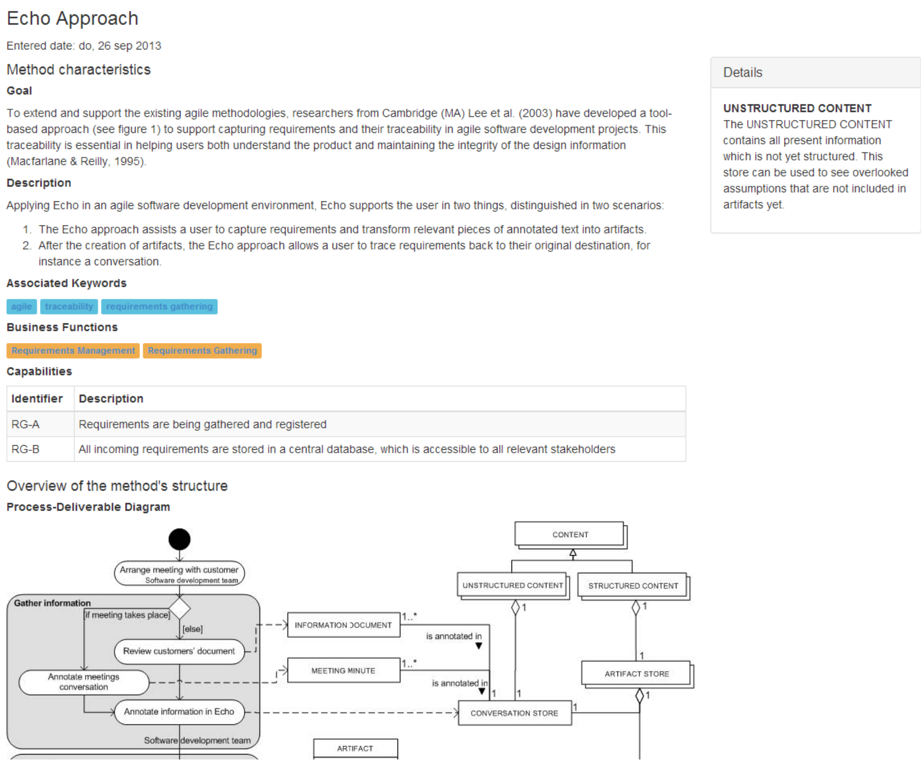


Fig. 3. Example Method Fragment Description

The results of the method assessment activity indicate the areas within the current process that are open for improvement. This is done in the form of a set of capabilities that are divided over various process areas, such as requirements prioritization or product roadmapping, and ordered based on an associated maturity level. This makes it possible to match these missing capabilities, or a subset of them, to existing methods that are stored in the method base.

The OME facilitates method discovery by providing tools to search the method base based on several filters. These filters include the focus area, the business function, the relevant capabilities, and keywords. The result is a set of method fragments that fit these filters. Each result can be expanded into a detailed description of the method fragment. The main components of these descriptions are a textual summary of the method fragment, a detailed description that can contain both text and illustrations, a list of the relevant situational factors and capabilities, a PDD, descriptions of the main activities and deliverables, and a list of reference. An example excerpt of a method description for the Echo approach [7] is shown in Figure 3.

Within the OME, the user is aided in this discovery process as much as possible. Elements within the PDD and items within the situational factor and capability lists are hyperlinks to more detailed descriptions. This makes it possible to obtain a quick overview, while allowing the user to go into more depth or to find related method fragments.

3.3 Scenario C: Method Improvement

An important goal of the OME is to assist during a process improvement effort. In many cases, there are various solutions available to solve a specific problem, such as requirements prioritization. Not all of these solutions are applicable to any given situation, and it is hard for the method engineer to determine which solutions are likely optimal. The knowledge available within the method base allows for an automated approach to selecting appropriate method fragments. This selection can be based on the structural aspects of the method fragment (activities, deliverables), required capabilities, and situational factors.

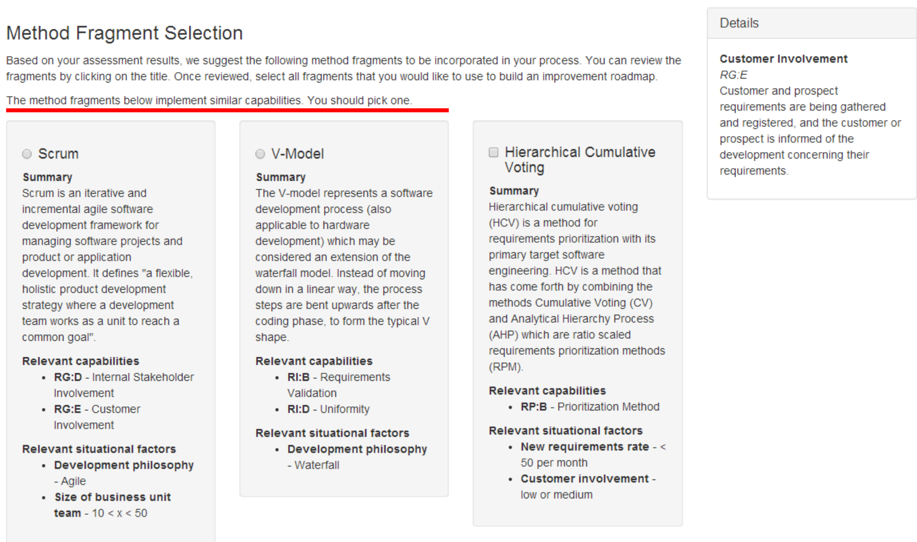


Fig. 4. Method Fragment Selection

The OME can propose a set of relevant method fragments based on the requirements of the user. It is possible to select method fragments that focus on one or more specific business functions, that implement capabilities up to a certain maturity level, or for the entire process. The selected fragments are presented to the user so that they can be reviewed using the tools described above (under method discovery). A prototype of the selection step is shown in Figure 4.

Once the user has selected the method fragments that he deems relevant, these fragments need to be assembled. In most cases, this means that they need to be integrated within the existing process. This activity can be partially performed automatically based on the structural aspects of the method fragments. In many cases, conflicts will arise during this assembly. For instance, multiple method fragments can include similar or incompatible deliverables. A report of these issues is presented to the user for further review.

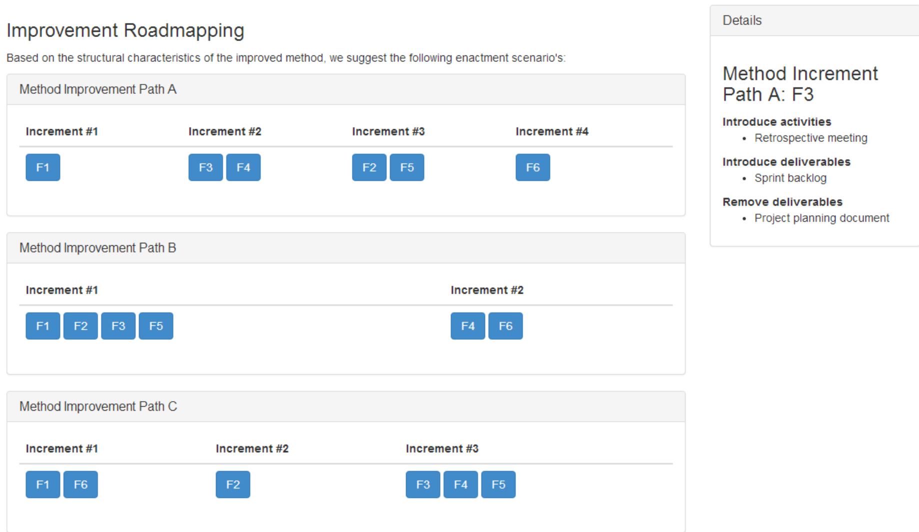


Fig. 5. Improvement Roadmapping

The overarching goal of the OME is to allow for incremental improvement. This is a very important characteristic, as the implementation of many process changes at the same time is often unrealistic and unfeasible within an organization. Process changes are always prone to resistance among employees, unforeseen complications, and changes within the environment. Therefor, the system generates a series of implementation plans [14]. These plans consist of a series of increments. Each increment consists of a set of small changes, such as inclusion or removal of activities and deliverables. Plans are generated based on a set of parameters, including the available resources and temporal constraints such as the need for a certain capability to be implemented within a certain amount of increments [14].

The generated plans are presented to the user as a set of timelines. Each timeline contains the proposed increments with a summary of their contents. Once again. the user can review these changes to gain a detailed understanding of their contents and impact (see FIgure 5 for an example of these timelines).

3.4 Scenario D: Method Enactment

The final activity within a process improvement effort, apart from the review, is the enactment of process changes. Although this is an activity that is mainly dependend on social and managerial aspects, it is possible to support parts of it through automated means. Within the context of the OME, enactment support focuses on the generation of templates and the automated migration of development tools.

3.5 Scenario E: Method Administration

All functionality within the OME is based on the contents of the method base. This method base consists of method fragments, capabilities, situational factors, situational factor effects, and experience reports. For the creation of method fragments, we employ specialized tools instead of developing functionality within the OME itself. MetaEdit+ [13] is used to model the process-deliverable diagrams, which can be annotated with relevant capabilities. Textual descriptions can be created with appropriate textual editors. References are stored in the BibTex format.

4 Discussion and Future Research

Both the conceptual design as well as the initial prototype of the OME have been validated in earlier studies [16, 15]. In its current iteration, the OME does not fully support all of the described scenarios. Development follows the layers as described in Figure 1 from bottom to top. The method base has been adequately implemented, as is functionality related to method discovery. The method assessment layer has also been realized, making it possible to assess current methods and link the results to method fragments proposal.

On the method improvement layer, we have realized partial planning functionality. This makes it possible to generate a set of improvement plans based on a goal method. However, more research is needed to incorporate the removal and replacement of method fragments, and to support more complex situations including improvement based on an existing method.

For the method enactment layer, no functionality has been implemented sofar. Techniques are currently under development to translate method changes into concrete enactment actions.

References

1. Baddoo, N.: De-motivators for software process improvement: an analysis of practitioners' views. Journal of Systems and Software 66, 23–33 (Apr 2003)

2. Bekkers, W., Spruit, M., van de Weerd, I., van Vliet, R., Mahieu, A.: A situational assessment method for software product management. In: Alexander, T., Turpin, M., van Deventer, J. (eds.) Proceedings of the European Conference on Information Systems. pp. 22–34. Pretoria, South-Africa (2010)
3. Diaz, M., Sligo, J.: How software process improvement helped Motorola. Software, IEEE 14(5), 75–81 (1997)
4. Hug, C., Front, A., Rieu, D.: A Process Engineering Method Based on Ontology and Patterns. In: Proceedings of the International Conference on Software and Data Technologies (2008)
5. Jarke, M., Gallersdörfer, R., Jeusfeld, M.A., Staudt, M., Eherer, S.: ConceptBase - A deductive object base for meta data management. Journal of Intelligent Information Systems 4(2), 167–192 (Mar 1995)
6. Karlsson, F., Å gerfalk, P.J.: MC Sandbox: Devising a tool for method-user-centered method configuration. Information and Software Technology 54(5), 501–516 (May 2012)
7. Lee, C., Guadagno, L., Jia, X.: An agile approach to capturing requirements and traceability. In: International Workshop on Traceability in Emerging Forms of Software Engineering (2003)
8. Nehan, Y.R., Deneckère, R.: Component-based Situational Methods: A framework for understanding SME. In: Situational Method Engineering: Fundamentals and Experiences. vol. 244, p. 161. Springer (2007)
9. Object Management Group: Meta Object Facility (MOF^{TM}). Tech. rep., Object Management Group (2011)
10. Pino, F.J., Pedreira, O., García, F., Luaces, M.R., Piattini, M.: Using Scrum to guide the execution of software process improvement in small organizations. Journal of Systems and Software 83(10), 1662–1677 (Oct 2010)
11. Ralyté, J.: Reusing scenario based approaches in requirement engineering methods: CREWS method base. In: International Workshop on Database and Expert Systems Applications (DEXA). pp. 305–309. No. Dexa 1999, Ieee (1999)
12. Ralyté, J., Rolland, C., Plihon, V.: Method enhancement with scenario based techniques. In: Proceedings of CAiSE. pp. 103–118 (1999)
13. Smolander, K., Lyytinen, K.: MetaEdit—a flexible graphical environment for methodology modelling. In: International Conference on Advanced Information Systems Engineering. pp. 168–193 (1991)
14. Vlaanderen, K., Dalpiaz, F., Brinkkemper, S.: Finding Optimal Plans for Incremental Method Engineering. Manuscript accepted for publication. In: International Conference on Advanced Information Systems Engineering (2014)
15. Vlaanderen, K., Dalpiaz, F., van Tuijl, G., Spruit, S., Brinkkemper, S.: Online Method Engine: a Toolset for Method Assessment, Improvement, and Enactment. Manuscript accepted for publication. International Journal of Information System Modeling and Design (IJISMD) (2014)
16. Vlaanderen, K., van de Weerd, I., Brinkkemper, S.: On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management. International Journal of Information System Modeling and Design (IJISMD) 3(4), 21 (2012)
17. Vlaanderen, K., van de Weerd, I., Brinkkemper, S.: Improving software product management: a knowledge management approach. International Journal of Business Information Systems (IJBIS) 12(1), 3–22 (2013)
18. van de Weerd, I., Brinkkemper, S.: Meta-modeling for situational analysis and design methods. In: Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, chap. III, pp. 35–54. Information Science Publishing (2008)
19. van de Weerd, I., Brinkkemper, S., Nieuwenhuis, R., Versendaal, J., Bijlsma, L.: Towards a Reference Framework for Software Product Management. 14th IEEE International Requirements Engineering Conference (RE'06) pp. 319–322 (Sep 2006)

SOVA – A Tool for Semantic and Ontological Variability Analysis

Nili Itzik and Iris Reinhartz-Berger

Department of Information Systems, University of Haifa, Israel
nitzik@campus.haifa.ac.il, iris@is.haifa.ac.il

Abstract. Variability analysis in Software Product Line Engineering (SPLE) utilizes various software-related artifacts, including requirements specifications. Currently, measuring the similarity of requirements specifications for analyzing variability of software products mainly takes into account semantic considerations. This might lead to failure to capture important aspects of the software behavior as perceived by users. In this paper we present a tool, called SOVA – Semantic and Ontological Variability Analysis, which introduces ontological considerations to variability analysis, in addition to the semantic ones. The input of the tool is textual requirements statements organized in documents. Each document represents the expectations from or the characteristics of a different software product in a line, while each requirement statement represents an *expected behavior* of that software product. The output is a feature diagram representing the variability in the input set of requirements documents and setting the ground for behavioral domain analysis.

Keywords: Software Product Line Engineering, Variability Analysis, Domain Analysis, Requirements Specifications, Ontology, Semantic Similarity

1 Introduction

As the complexity and variety of software products increased, the need to reuse software-related artifacts became very important. Software Product Line Engineering (SPLE) suggests an approach to systematically reuse artifacts, such as requirements specifications, design documents and source code, among different, yet similar, software products [3], [14]. Such reuse of artifacts often raises a significant challenge of variability management. *Variability* in this context can be defined as “the ability of an asset to be efficiently extended, changed, customized, or configured for use in a particular context” [7].

Viewing software requirements as the drivers of different development activities and methods, several studies have suggested using requirements specifications for variability analysis of software products. In these studies, requirements are operationalized or realized by features, and variability is mainly represented as *feature diagrams* – tree or graph structures that describe the characteristics of a software product line and the relationships and dependencies among them [8]. The current studies commonly apply only semantic similarity metrics, which focus on similarities of terminology, in order to identify and analyze variability. As we will elaborate later, using only semantic considerations might lead to failure to capture important aspects of the software behavior, such as its triggers, pre-conditions, and post-conditions.

In [16], we suggest combining semantic and ontological considerations for calculating similarity. In particular, a behavior is described in terms of the initial state of a system before the behavior occurs, the external events that trigger the behavior, and the final state of the system after the behavior occurs. We use semantic metrics to evaluate the similarity of related behavioral elements and utilize this similarity to analyze variability. To support this approach, we have developed a tool, called SOVA – Semantic and Ontological Variability Analysis. This tool gets requirements documents written in plain text. Each document represents a different software product in the line and is divided into requirements statements. Each requirement statement, which may be composed of several sentences, reflects a use case, a user story, or any unit that represents a single expected or existing behavior of a software product. The variability of requirements is then analyzed, yielding a feature diagram. The resultant feature diagrams are behavior-driven and set the ground for behavioral domain analysis.

The rest of this paper is structured as follows. Section 2 reviews related work, exemplifying limitations of current approaches. Section 3 presents the main processes of the approach and their support in the SOVA tool. Finally, Section 4 summarizes and refers to future development plans.

2 Related Work

In the context of analyzing software products variability, different studies have suggested ways to use textual requirements to generate variability models, such as feature diagrams or Orthogonal Variability Models (OVM) [14].

In [19], a tool, named ArborCraft, is presented. This tool creates feature diagrams by grouping similar requirements using a hierarchical agglomerative clustering algorithm and semantic similarity measures – Latent Semantic Analysis (LSA) [10]. Feature variants are then identified using a Requirements Description Language and semantic considerations. In [4-5], publicly available repositories of product descriptions are utilized. Based on these repositories and the conditional probabilities between features occurrences, a probabilistic feature diagram is created using an incremental diffusive clustering algorithm. In [13], a semi-automatic method for constructing OVM diagrams is introduced. This method extracts functional requirements profiles (FRPs), represented as "verb-direct object" pairs, using expert knowledge and linguistic clues. The variability model is created using heuristic rules, such as: “If diverse values are identified for a case, then alternative choice(s) should be made.”

All the above methods employ only semantic considerations. In particular, they may result with high similarity values for requirements that use similar terminology, even if the pre-conditions, the triggers, and the post-conditions of the corresponding behaviors are different. For example, the requirements “The system should be able to report on any user update activities" and "Any user should be able to report system activities" may result in a very high value of semantic similarity, since both refer to "system", "user", and "report". In fact, LSA [10] results in a similarity value of 1 for these requirements, implying that their semantic meanings are identical. However, these requirements are quite different: the first requirement represents behavior that is internal and likely aims at detecting suspicious user update activities. The second requirement, on the other hand, represents a behavior triggered by an external user who intends to report his/her system activities.

Another limitation of current studies is that they take into consideration the full text of a requirement statement. Such statements might include aspects (e.g., intermediate outcomes) that are less or not relevant for analyzing variability from an external perspective of a user or a customer. Such a view of the *expected behaviors* of software systems is important for reaching different reuse decisions, e.g., when conducting feasibility studies, estimating software development efforts, or adopting SPLE.

To overcome the above limitations, we proposed in [16] to combine semantic and ontological considerations when calculating similarity and analyzing variability. We further demonstrated that our approach outperforms LSA when examining the similarity of functional requirements. Here we present the tool we have developed to support that approach. The tool is named SOVA – Semantic and Ontological Variability Analysis.

3 The SOVA Tool

Fig. 1 presents the main processes supported by the SOVA tool, namely requirements parsing, behavioral similarity calculation, and feature diagram creation. Next we elaborate on each process and its support in the tool. Additional material can be found at http://mis.hevra.haifa.ac.il/~iris/research/SOVA/.

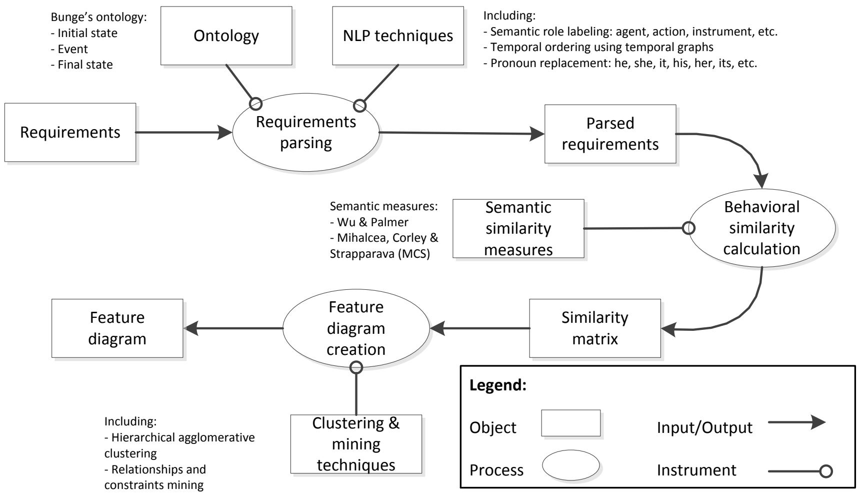


Fig. 1. An overview of the processes and flows supported by the SOVA tool

3.1 Requirements Parsing

During the first step, the input requirements are parsed. This is done by two main instruments: natural language processing (NLP) techniques and an ontological model.

First, a semantic role labeling (SRL) approach [6] is used to associate the parts of a requirement statement with their specific semantic roles. Five semantic roles are currently supported due to their special importance to requirements in general and functional requirements in particular: (1) Agent – Who performs? (2) Object (a.k.a. Patient) – On what objet is it performed? (3) Instrument – How is it performed? (4) Temporal modifier (AM-TMP) – When is it performed? And (5) Adverbial modifier (AM-ADV) – In what conditions is it performed? A sixth label – Action – is handled to answer the question: What is performed? This label holds the sentence's predicate or verb.

Considering those labels and applying temporal order [11] and coreference resolution[1] [15] techniques, the tool identifies behavioral vectors, each representing an action or a pre-condition. Using concepts taken from Bunge's ontological model [1-2], the behavioral vectors are then classified into initial states that represent pre-conditions of the behavior, external events that trigger the behavior, and final states that represent post-conditions or outputs of the behavior. These three "types" of behavioral elements (namely, initial states, external events, and final states) were suggested in [17-18] for defining an external view of behavior. The classification of the vectors to these behavioral elements is mainly done by analyzing the *agent* and the *action* parts of the vectors and using the temporal order of the vectors [16].

The screenshot presented in Fig. 2 exemplifies the outcome of the parsing requirements activity. The field at the top of this screen enables choosing a particular requirements file and browsing its requirements statements (in the middle part of this screen). Each requirement statement includes one or more sentences. Each sentence appears in a separate row, where the number to its left indicates the requirement to which it belongs. Requirement 2, for example, is composed of two sentences. Choosing a particular sentence displays the parsing of the entire requirement to which the sentence belongs in the bottom part of this screen. The second requirement in Fig. 2, for example, is parsed into three behavioral vectors. The first vector is classified as an initial state, since it represents a pre-condition (labeled as a temporal modifier). The second vector, representing a login operation, is classified as an external event, since it is performed by an external agent – the librarian. Finally, the third vector is classified as a final state, as it describes an internal operation performed by the system after the librarian logins.

During the parsing process, the tool further supports interactions with the user, namely, a requirements engineer or a domain analyst. In particular, the user can edit the ontological class, change the order of the parsed behavioral vectors, update the original requirements, and view the semantic role labeling output (the SRL button).

---

[1] Coreference resolution replaces pronouns (e.g., he, she, and it) with their anaphors (i.e., the nouns to which they refer).

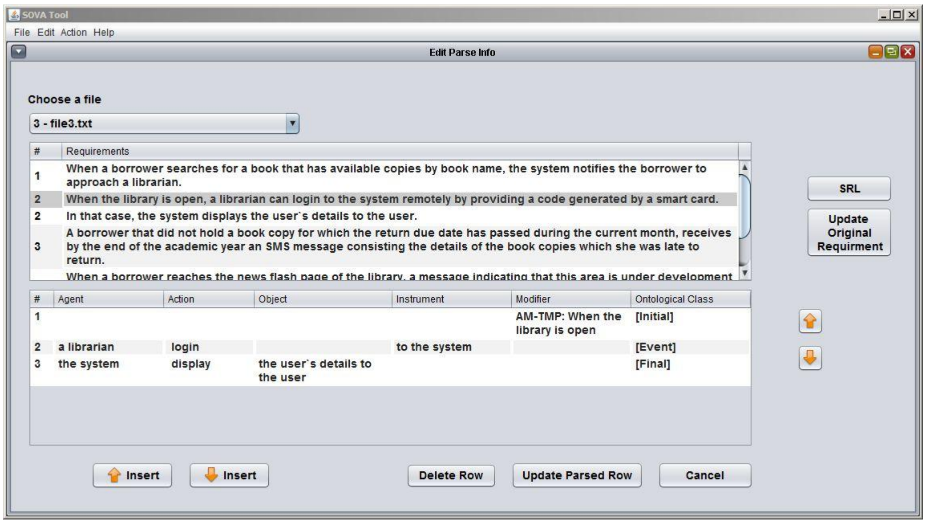


Fig. 2. A screenshot of the requirements parsing outcome

3.2 Behavioral Similarity Calculation

In the second process, the behavioral similarity of each pair of requirements (either from the same document or from different documents) is calculated. The behavioral similarity is the weighted average of the semantic similarities of their behavioral vectors. In other words, the behavioral similarity is the weighted average of the semantic similarities of their initial states, external events, and final states. For calculating the semantic similarities of the behavioral elements different semantic measures can be used. Here we use MCS [12] to measure phrases' similarity and Wu and Palmer [20] to measure words' similarity. The user can further set the weights for agents, actions, objects, and instruments similarities. Perceiving agents and actions as the dominant components in behavioral vectors similarities, Fig. 3 exemplifies the outcome of the behavioral similarity calculation process in SOVA, using 0.3, 0.4, 0.2, and 0.1 for weighting agents, actions, objects, and instruments, respectively. The screen displays (in the right side) the initial state, external event, final state, and overall similarities for each pair of requirements in the source files. The overall similarity is calculated using initial state, external event and final state weights of 0.2, 0.3, and 0.5, respectively, perceiving the final state as the most influencing factor on the overall similarity.

In Fig. 3, for example, the first pair of requirements (the ninth requirement in the first input file and the forth requirement in the third input file) represents different cases (initial states) and responses (final states), but similar interactions (external events) in which someone (visitor or borrower) reaches the new flash page of the library. The requirements in the second row represent very similar behaviors, which differ only in their agents (users vs. librarians). Finally, the requirements in the third row represent completely different behaviors.

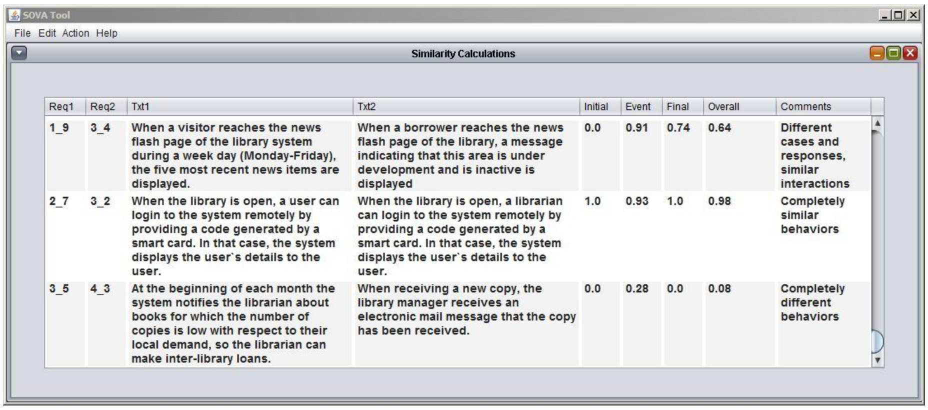


Fig. 3. A screenshot of the behavioral similarity calculation outcome

3.3 Feature Diagram Creation

In the third process, we use the calculated similarity values in order to create a feature diagram that represents the variability found in the input requirements documents. To this end, we utilize a hierarchical agglomerative clustering algorithm. This algorithm starts with putting each requirement in a separate cluster. In each iteration, the algorithm merges the closest clusters, namely, clusters whose average requirements' similarities is the highest. The output of this algorithm is a binary tree of clusters. To better represent the analyzed variability, another pass is performed to flatten sub-trees whose similarities are alike. To demonstrate this pass, consider the schematic tree in the left side of Fig. 4. The leaves of this tree represent requirements (or actually clusters with single requirements), numbered 1 to 5, while the inner nodes represent clusters with several requirements. Each inner node exhibits its identity (e.g., C1:2_4) and the overall similarity of the constituting requirements. Note that the sub-tree whose root is C1:2_4 includes very similar requirements, namely R1, R2, and R4. Therefore, in the flatten tree (in the right side of the figure), the three requirements have the same parent. In contrast, the node C3_1:2:4 holds a requirement, R3, which is quite different from the other related requirements, R1, R2, and R4. Thus, grouping the four requirements together is unjustified. Instead R3 and C1:2_4 become siblings in the flatten tree.

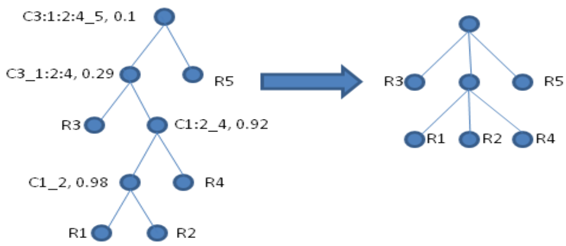


Fig. 4. Illustration of flattening the clustering outcome in the feature diagram creation stage

Optionality as well as OR- and XOR-grouped features are deduced examining the appearance of the different requirements in the input requirements documents. The final output is presented in featureIDE format. FeatureIDE is an eclipse plug-in that supports different phases of the feature-oriented software development [9]. It is user friendly. In particular, the feature diagrams can be presented horizontally or vertically, the requirements can be presented as description of leave nodes, and the diagrams can be exported to a variety of feature diagram formats.

The SOVA tool enables generating feature diagrams according to different behavioral views, namely, considering only the similarity of the initial states, the external states, the final states, or the overall behaviors. Thus, and as opposed to existing approaches and tools, the variability of the requirements can be analyzed from different perspectives. For example, considering only the similarity of final states may provide an output-driven variability perspective, while considering the external events provides a functional variability perspective.

4 Summary and Future Work

We presented a tool, named SOVA – Semantic and Ontological Variability Analysis – that supports identifying and analyzing behavioral variability of software products based on requirements specifications. The tool combines semantic and ontological considerations through a three stage process that includes parsing the requirements using NLP techniques and Bunge's ontological model, calculating the behavioral similarity of software requirements using semantic measures, and generating feature diagrams using a hierarchical agglomerative clustering algorithm. All these processes are done automatically and the user is only required to set weights for the different semantic similarities.

In the future, we intend to extend the tool support in several ways. First, we intend to involve the user throughout the process and to allow him/her to provide intermediate feedback which will be taken into consideration in the following stages. Second, we intend to derive state variables from intermediate states and not just from initial and final states. These state variables may further help identify the commonality and variability of software products by refining the external view. Finally, we intend to handle requirements statements that represent "swarms" of behaviors (including branches and loops) and not just single ones. This will enable us to analyze relationships between requirements and not just individual requirements.

References

1. Bunge, M. (1977). Treatise on Basic Philosophy, vol. 3, Ontology I: The Furniture of the World. Reidel, Boston, Massachusetts.
2. Bunge, M. (1979). Treatise on Basic Philosophy, vol. 4, Ontology II: A World of Systems. Reidel, Boston, Massachusetts.
3. Clements, P. and Northrop, L. (2001). Software Product Lines: Practices and Patterns. Addison-Wesley.

4. Davril, J. M., Delfosse, E., Hariri, N., Acher, M., Cleland-Huang, J., and Heymans, P. (2013). Feature model extraction from large collections of informal product descriptions. The 9th Joint Meeting on Foundations of Software Engineering, pp. 290-300.
5. Dumitru, H., Gibiec, M., Hariri, N., Cleland-Huang, J., Mobasher, B., Castro-Herrera, C., and Mirakhorli, M. (2011). On-demand feature recommendations derived from mining public product descriptions. 33rd IEEE International Conference on Software Engineering (ICSE'11), pp. 181-190.
6. Gildea, D. and Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. Computational Linguistics 28 (3), pp. 245-288.
7. Jaring, M. (2005). Variability engineering as an Integral Part of the Software Product Family Development Process, Ph.D. thesis, The Netherlands.
8. Kang, K. C., Cohen, S. G., Hess, J. A., Novak, W. E., and Peterson, A. S. (1990). Feature-oriented domain analysis (FODA) – feasibility study. Technical report no. CMU/SEI-90-TR-21). Carngie-Mellon University, Pittsburgh.
9. Kastner, C., Thum, T., Saake, G., Feigenspan, J., Leich, T., Wielgorz, F., and Apel, S. (2009). FeatureIDE: A tool framework for feature-oriented software development. 31st IEEE International Conference on Software Engineering (ICSE'09), pp. 611-614.
10. Landauer, T. K., Foltz, P. W., and Laham, D. (1998). Introduction to Latent Semantic Analysis. Discourse Processes, 25, pp. 259-284.
11. Mani, I., Verhagen, M., Wellner, B., Lee, C. M., and Pustejovsky, J. (2006). Machine learning of temporal relations. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, pp. 753-760.
12. Mihalcea, R., Corley, C., and Strapparava, C. (2006). Corpus-based and knowledge-based measures of text semantic similarity. The 21st national conference on Artificial intelligence (AAAI'2006), Vol. 1, pp. 775-780.
13. Niu, N. and Easterbrook, S. (2008). Extracting and modeling product line functional requirements. In the 16th IEEE International Requirements Engineering conference (RE'08), pp. 155-164.
14. Pohl, K., Böckle, G., and van der Linden, F. (2005) Software Product-line Engineering: Foundations, Principles, and Techniques, Springer.
15. Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., and Manning, C. (2010). A Multi-Pass Sieve for Coreference Resolution. The conference on Empirical Methods in Natural Language Processing (EMNLP'10), pp. 492-501.
16. Reinhartz-Berger, I., Itzik, N., and Wand, Y. (2014). Analyzing Variability of Software Product Lines Using Semantic and Ontological Considerations Proceedings of the 26th international conference on Advanced Information Systems Engineering (CAiSE'14), LNCS 8484, pp. 150-164.
17. Reinhartz-Berger, I., Sturm, A., and Wand, Y. (2013). Comparing Functionality of Software Systems: An Ontological Approach. Data & Knowledge Engineering87, pp. 320-338.
18. Reinhartz-Berger, I., Sturm, A., and Wand, Y. (2011). External Variability of Software: Classification and Ontological Foundations. The 30th International Conference on Conceptual Modeling (ER'2011), LNCS 6998, pp. 275-289.
19. Weston, N., Chitchyan, R., and Rashid, A. (2009). A framework for constructing semantically composable feature models from natural language requirements. In Proceedings of the 13th International Software Product Line Conference, pp. 211-220.
20. Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. The 32nd annual meeting on Association for Computational Linguistics, pp. 133-138.

A Prototype Tool for Modeling and Analyzing Security Requirements from A Holistic Viewpoint

Tong Li, Jennifer Horkoff, John Mylopoulos

University of Trento, Trento, Italy
{tong.li,horkoff,jm}@disi.unitn.it

Abstract. Security breaches in large socio-technical systems cost billions. Many breaches can be attributed to the piecemeal security design, leaving parts of the system vulnerable while others are over-protected. We advocate holistic security design, and have introduced techniques to support security analysis across multiple layers. This paper presents MUSER, a prototype that assists security requirements analysts dealing with security requirements from a holistic viewpoint based on a three-layer framework. Our prototype analyzes security requirements and related security mechanisms in business layer, application layer, and physical layer. The prototype captures the influences of security mechanisms, which one layer enforces on the other layers, and supports deriving holistic security solutions that tackle security concerns in all layers. We demonstrate the usage of MUSER via a smart grid scenario.

Keyword: Security Requirements · Goal Model · Multilayer · Socio-Technical System · Demo Tool

1 Introduction

Socio-technical systems (STSs) are organizational systems consisting of people, business processes, software applications, and hardware components. As the complexity of STS increases over time, a growing number of security breaches are reported [1].

A common theme for many of these breaches is that security solutions are dealt with in a piecemeal fashion, in which security analysis carried out in one part of the system does not take into account security designs in other parts. For example, when designing an encryption function, a smart meter application can either implement encryption by itself or depend on an external component implemented in a specialized chip. If the system is viewed in a piecemeal fashion, for example, focusing only on the software issues, these two security mechanisms deliver the same function. However, these two alternatives have different influences on requirements of the physical devices, as the latter one requires that the related hardware device should not be accessed by non-authorized people [2], a factor often not accounted for during physical design of the system. As reported in [2], attacks that exploiting this vulnerability have been done by bus-snooping.

In this paper, we present MUSER (MUltilayer SEcurity Requirements analysis tool), a prototype that supports analyzing security requirements of STSs from a holistic point of view, implementing our technique proposed in [3]. Our approach structures STSs into three layers, namely, business layer, application layer, and physical layer. By carrying out analysis both inside one layer and across layers, the approach is intended to generate holistic security solutions for STSs with regard to stakeholder's high-level security needs. To assist the security analysis, our prototype provides the following features: 1) multilayer requirements modeling, 2) hierarchical security requirements refinement, 3) identification of critical security requirements, 4) generation of potential security mechanisms, 5) cross-layer security influence analysis.

In the reminder of this paper, we first describe our analysis approach in Sec. 2, according to which we design our prototype. Next, we introduce the architecture and functionality of our prototype in Sec. 3, and illustrate the utility of the prototype in Sec. 4. Finally, Sec. 5 concludes the paper.

2 Three-Layer Security Requirement Analysis Approach

Our previous work [3] has proposed a goal-oriented three-layer analysis framework, which analyzes security requirements of STSs. This approach adopts concepts from existing well-established goal-oriented modeling languages, such as Techne [4] and i* [5]. To support the three-layer framework, we specialize the concepts *Goal* and *Task* into layer-specific concepts. For example, *Task* is specialized as *Business Process Activity* in the business layer, while it is specialized as *Application Function* in the application layer. In addition, we define *Security Goal* as a special *Softgoal*, and assign it with formal semantics via four dimensions: *Importance*, *Security Attribute*, *Asset*, and *Interval*. For example, a security goal *Medium Integrity[customer information, measure energy consumption]* describes a security requirement "protecting integrity of customer information during the executing period of measure energy consumption to a medium degree".

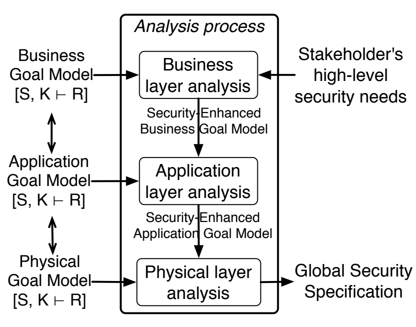


Fig. 1: Overview of the three-layer analysis framework

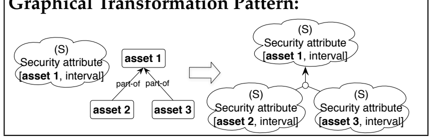
Asset-based Refinement Rule
Content: If an asset consists of sub-assets, a security goal that concerns this asset could be refined to more detailed ones, which focus on the sub-assets.
Datalog Rule: $and\_refined\_sec\_goal(IMP, SA, AS2, INT, SG) :- sec\_goal(SG), importance(IMP), sec\_attribute(SA), asset(AS2), interval(INT), part\_of(AS2, AS1), asset(AS1), has\_properties(SG, IMP, SA, AS1, INT).$
Graphical Transformation Pattern: 

Table 1: Asset-based refinement rule

Our approach analyzes the requirements problem [6] within each of the three layers respectively. In this sense, each layer has its own requirements

R, which are satisfied by its own specifications S, under layer-specific domain assumptions K, i.e. $S, K \vdash R$. In particular, specifications in one layer dictate requirements in lower layers, indicated in Fig. 1. By capturing the cross-layer interactions and analyzing the requirements problem within each individual layer, our approach aims at dealing with security requirements of STSs from a holistic viewpoint. Taking stakeholder's high-level security requirements and layer-specific goal models as input, the approach analyzes detailed security requirements throughout the three layers and finally produces holistic security solutions, which tackle security issues in all three layers (Fig. 1).

The approach includes 23 transformation rules, which guide the corresponding security analysis tasks. Specifically, the rules support refining, simplifying and operationalizing security goals within individual layers, as well as transferring security concerns across layers. All of these rules have been implemented in Datalog [7], which can be automatically inferred with corresponding inference engines, such as DLV [8]. Each rule has been documented in a template, consisting of three sections: *Content*, *Datalog Rule* and *Graphical Transformation Pattern*. Table 1 shows an example regarding the asset-based refinement rule, which refines a security goal via its asset dimension. The full list of transformation rules is available online [1].

3 MUSER Prototype Tool

MUSER (MUltilayer SEcurity Requirements analysis tool) is a prototype, which is designed to support the application of the approach summarized in Sec. 2. The prototype is a Java-based program, which is developed on the top of a specialized and powerful diagramming application *OmniGraffle* [2]. Fig. 2 shows the architecture of the prototype, which consists of four components: *control*, *view*, *model*, and *inference*. We introduce them respectively as below.

Control Component controls the logic of the whole prototype and coordinates other components to deliver inference functions to users. When receiving users' inference requests, it imports related models from the view component, generates a formal model specification in terms of text files, and calls the inference component to carry out corresponding reasoning tasks. According to the reasoning results returned by the inference component, the control component updates related model information and reflects them on the view component.

View Component supports users with graphic modeling, as well as shows graphical inference results to users. The main requirements for this component include: 1) support goal-oriented modeling and allow customized notations; 2) support multilayer modeling, i.e. modeling in different views; 3) be connected with inference component to support reasoning. Although there are a number of available goal modeling tools, such as Open OME, STS-ml [3], none of them can support

[1] http://goo.gl/Pd0TGw
[2] http://www.omnigroup.com/omnigraffle
[3] http://istar.rwth-aachen.de/tiki-index.php?page=i*+Tools

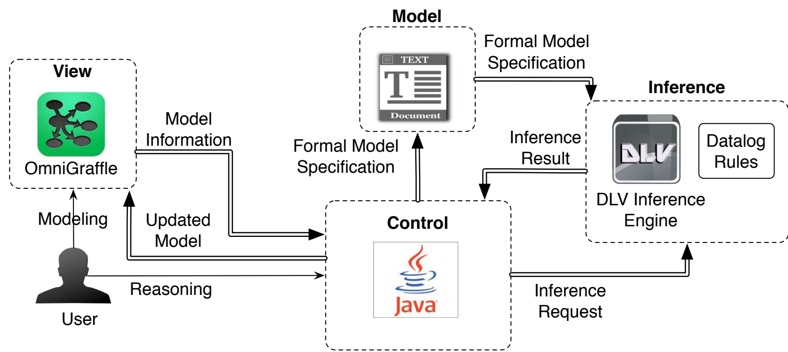


Fig. 2: An overall architecture of the prototype

our multilayer goal modeling and reasoning. We choose a specialized modeling application *OmniGraffle* as the view component of our prototype, meeting all requirements. Particularly, this application has many useful modeling features, such as automatic layout, outline view, and various export formats. As we have defined a number of interfaces for the view component, through which it interacts with the control component, the view component can be replaced by other applications that comply with the interfaces.

Inference Component implements the security analysis rules proposed in our approach and automates corresponding analysis tasks. All the rules are implemented in terms of Datalog rules and facts. Particularly, we leverage DLV inference engine [8] to carry out inference actions. This component receives requests from the control component, and returns inference results back afterwards.

Model Component is responsible for storing the formal models, which are required by the inference component. We use text files for storage, which fit our current needs. If the analysis involves very complex and large models in the future, we will replace the text files with specialized database applications.

3.1 Functionality

Building on the above architecture, we implement a number of functions, which assist the security analysis tasks. As shown in Fig. 3, within one single layer, we first refine and simplify security goals to identify concrete and critical ones. Then, the critical security goals are operationalized into possible security mechanisms and left to security analysts to select. After that, we transfer security concerns downward to its lower layer and iteratively carry out all above security analysis there. The main features of the tool are as follows:

– *Security Goal Refinement:* As a coarse-grained security goal is normally more difficult to analyze and operationalize than a fine-grained one, security analysts need to refine an abstract security goal into sub-goals. The prototype

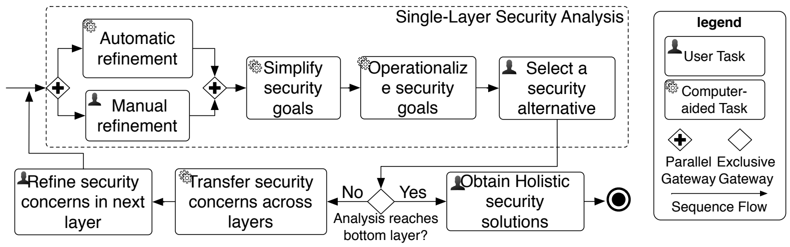


Fig. 3: Detailed security analysis procedure

supports automatically refining a security goal via any of these three dimensions: *security attribute*, *asset*, and *interval*. Particularly, it provides two ways to automate this analysis. First, it supports a single step of refinement with regard to a particular dimension. Secondly, the prototype could simulate an exhaustive analysis, which explores all possible refinements for the target security goals, to cover all related security goals. Not surprisingly, this simulation could produce a very huge refinement trees, such as shown in Fig. 5. Thus, this analysis has to rely on simplification analysis for filtering non-critical security goals.

– *Security Goal Simplification:* When dealing with a large number of security goals, analysts may not have enough time to go through each of them to determine which is critical and requires further treatments. To release analysts from scrutinizing all security goals, our prototype supports automatic identification of critical security goals, which takes into account the *applicability* and *risk level* of security goals. For example, if a security goal concerns *data confidentiality* of *energy production data* during the execution of *measure energy consumption*, which does not involve with the *energy production data*, then this security goal is not applicable and is further identified as non-critical. In addition, for an applicable security goal, if its risk level is *low* or *medium*, it is identified as non-critical.
– *Security Goal Operationalization:* To effectively achieve critical security goals, our prototype assists analysts in designing security mechanisms based on existing security patterns [9]. A security pattern consists of a security attribute and a security mechanism that is supposed to satisfy that security attribute. When operationalizing a critical security goal, our prototype identifies all matched security patterns with regard to security attributes and generates corresponding security mechanisms for the critical security goal.
– *Cross-Layer Analysis:* To analyze security requirements from a holistic view, the influences of security analysis results in one layer should be propagated to layers below. This analysis takes into account designed security mechanisms and untreated critical security goals in the upper-layer, and generates corresponding security concerns in the lower-layer. For example, an untreated security goal in the business layer, which concerns data confidentiality, is refined into two sub-goals in the application layer that target corresponding applications according to the cross-layer analysis rules [3].

4 Demonstration

We demonstrate the utility of our prototype by holistically analyzing the security requirements of a smart grid example. Particularly, we focus on the real-time pricing scenario.

Real-time Pricing Scenario: *To continuously provide energy to customers, the energy provider needs to balance the load on the power grid to avoid blackouts. To this end, the provider applies a real-time pricing strategy, which adjusts energy price according to current load of the power grid. This strategy requires periodically collecting customer's energy consumption data, based on which a new price is calculated. In this scenario, the energy provider highly relies on the integrity of the energy consumption data.*

According to the detailed analysis procedure shown in Fig. 3, we use our prototype to analyze the security requirements for the above scenario. Our analysis technique will apply to each layer, here we present the security analysis within the business layer, with steps as follows:

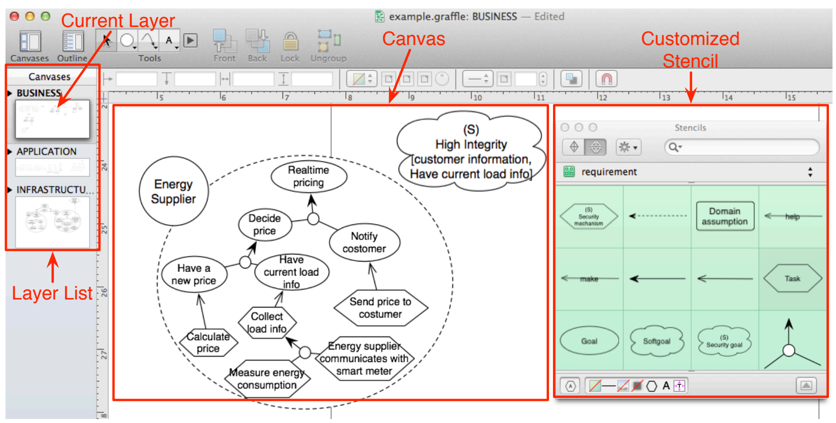


Fig. 4: Graphical modeling interface

1. *Building An Initial Requirements Model:* We first build the requirements model within the business layer, including the high-level security requirements. Fig. 4 shows the graphical modeling interface, which consists of several key components that are highlighted in rectangles. The *Layer List* shows three canvases that are intended for modeling requirements in the three layers. A customized stencil is shown in the bottom part of Fig. 4. Elements in the stencil are used to draw the requirements models, which is presented in the center of the canvas. Due to space limitation Fig. 4 only shows a part of the requirements model.
2. *Refine High-Level Security Goals:* Given the high-level security goal *High Integrity[customer information, Have current load info]* (Fig. 4). We use the prototype to adopt an exhaustive refinement strategy to cover all potential security concerns introduced by this security goal. Not surprisingly, this strategy results in a large refinement tree, which contains 72 potential security goals, shown in Fig. 5. Limited by space, we replace the content of each

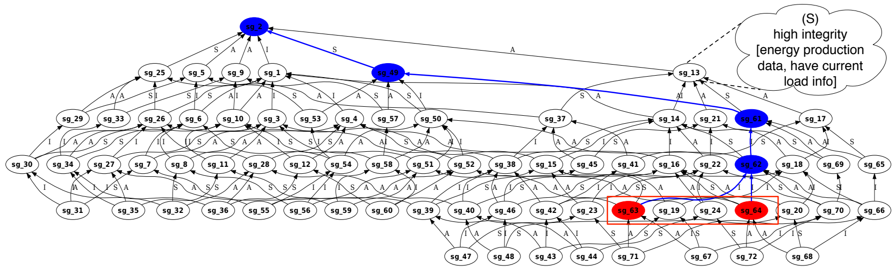


Fig. 5: Example of exhaustive refinement

security goal with its id, and only show the details of one security goal in the upper right corner.

3. *Simplify Security Goals:* As the exhaustive refinement analysis generates a large refinement tree of security goals, which are almost impossible to be analyzed manually, we use our prototype to automatically identify critical security goals. As highlighted by the rectangle in Fig. 5, 2 out of 72 security goals (the filled elements) are identified as critical. Based on these critical security goals, the prototype follows a bottom-up algorithm to calculate the best refinement path, which refines the top-level security goal into the critical ones with minimum steps (the filled elements above the rectangle in Fig. 5). After this simplification analysis, our prototype can generate a simplified security goal model, which contains only the critical security goals and the corresponding best refinement paths.
4. *Operationalize Security Goals:* Facing the two critical security goals, our prototype automatically generates four potential security mechanisms for each of them according to the security patterns [9]. Fig. 6 shows the operationalization of one critical security goal. It is worth noting the security mechanisms suggested by the security patterns may not fit for the security goals in particular system settings, so further judgments from security analysts are required. Furthermore, the prototype calculates all of the alternative security solutions which treat all critical security goals. In our example, the prototype totally generates 25 alternative solutions. Then the security analyst should choose one solution among the alternatives.
5. *Transfer Security Concerns:* After security analysts determine specific security solutions in the business layer, our prototype supports automatically transferring the related security concerns to the application layer, i.e. generating new security requirements with regard to the result of security analysis in this layer. Fig. 7 shows an example of such transfers with regard to the security mechanism *Cryptographic control* selected in the business layer. Note that choosing different security solutions in the business layer will result in different security requirements in the application layer.
6. *Iterative Security Analysis:* After transferring security concerns, we can iteratively carry out security analysis in the application layer and the physical layer, where our prototype automates tasks as in the business layer. Finally,

we will derive holistic security solutions by synthesizing security solutions selected in each individual layer.

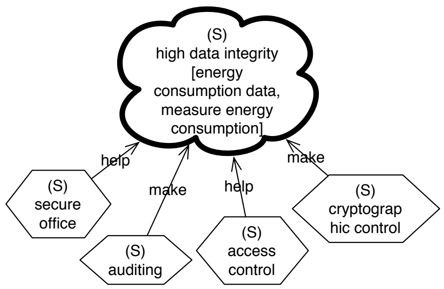


Fig. 6: Example of operationalization

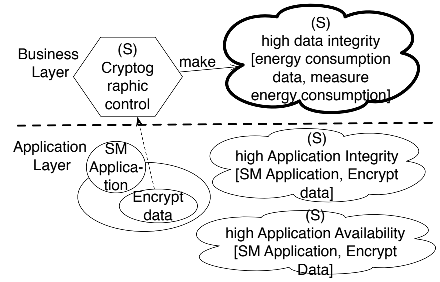


Fig. 7: Example of cross-layer analysis

5 Summary and Future Work

In this paper, we present MUSER, a prototype for analyzing security requirements of STSs from a holistic viewpoint. This prototype is designed to implement our proposed three-layer analysis approach [3], which supports security analysis both within one layer and across layers. In the future, we plan to carry out empirical experiments with our prototype to evaluate its usability and performance, and make further improvements.

Acknowledgements This work is supported by ERC advanced grant 267856, titled "Lucretius: Foundations for Software Evolution".

References

1. Tony Flick and Justin Morehouse. *Securing the smart grid: next generation power grid security*. Elsevier, 2010.
2. M Carpenter, T Goodspeed, B Singletary, E Skoudis, and J Wright. Advanced metering infrastructure attack methodology. *InGuardians white paper*, 2009.
3. Jennifer Horkoff Tong Li. Dealing with security requirements for socio-technical systems: A holistic approach. In *the 26th International Conference on Advanced Information Systems Engineering (CAiSE'14)*. Accepted, 2014.
4. I.J. Jureta, A. Borgida, N.A. Ernst, and J. Mylopoulos. Techne: Towards a new generation of requirements modeling languages with goals, preferences, and inconsistency handling. In *Proc. of RE'10*, pages 115–124, 2010.
5. Eric Yu. Towards modelling and reasoning support for early-phase requirements engineering. pages 226–235. IEEE Computer Soc. Press, 1997.
6. Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. *ACM Trans. Softw. Eng. Methodol.*, 6(1):1–30, 1997.
7. Thomas Eiter, Georg Gottlob, and Heikki Mannila. Disjunctive datalog. *ACM Transactions on Database Systems (TODS)*, 22(3):364–418, 1997.
8. Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)*, 7(3):499–562, 2006.
9. Riccardo Scandariato, Koen Yskout, Thomas Heyman, and Wouter Joosen. Architecting software with security patterns. Technical report, KU Leuven, 2008.

Visual and Ontological Modeling Support for Extended Enterprise Models

Sagar Sunkle and Hemant Rathod

Tata Research Development and Design Center
Tata Consultancy Services
54B, Industrial Estate, Hadapsar
Pune, 411013 INDIA
{sagar.sunkle,hemant.rathod}@tcs.com

Abstract. Modern enterprises have to face changes brought on by multiple change drivers like evolving market conditions, technology obsolescence and advance, and regulatory compliance among others. Enterprises need to create and use both descriptive and prescriptive models such that prescriptive models leverage the descriptive models to operationalize optimum strategies in response to change. This paper presents a visual model editor and ontological support for aforementioned kinds of models of enterprise. The editor enables modeling a) motivations behind and goals in response to change, b) the AS-IS state of enterprise, c) possible TO-BE states, and d) operationalization model that captures paths from AS-IS to desired TO-BE states. The analyses required are carried out using ontological representation.

Keywords: Enterprise Architecture Modeling, Intentional Modeling, Motivational Modeling

1 Introduction

For enterprises to respond to changes in an efficient and effective manner requires complete understanding of AS-IS architecture, possible TO-BE architectures, a way to evaluate TO-BE architectures based on some criteria, and an operationalization path from AS-IS architecture to the desired TO-BE architecture.

In this regard, earlier we investigated an approach in which intentional modeling was treated as a enterprise problem solving technique [1]. We represented AS-IS enterprise architecture (EA) models using Archi [2] and intentional models for TO-BE EA using OpenOME [3]. We carried out the evaluation of alternatives in OpenOME. Only a single alternative from amongst the optimum alternatives was then materialized over the AS-IS enterprise model. The AS-IS architecture model coupled with modification and addition of elements and relations would indicate a specific TO-BE architecture. This TO-BE architecture captured the intentional alternative found to be optimum.

As we applied this procedure to several real world case studies, we found that it had shortcomings that we enlist below, which became evident when we started modeling large real world enterprise models-

1. For really large enterprise models, keeping the EA and intentional modeling concerns in sync in two different modeling tools became nearly impossible as models grew in size.
2. In our original approach, the traceability between intentional models and elements of TO-BE architecture was not preserved. We had to adopt an ad-hoc process in enacting desired TO-BE architecture model presuming that concerns expressed in intentional models are represented adequately in TO-BE architecture model.

In accordance with the pointers mentioned above, we extend Archi to enable integrated visual modeling of EA models as descriptive models and intentional and motivational models as prescriptive models. We also extend the EA ontology we presented in [4] to carry out various analyses required in terms of computing prescriptive courses of action from AS-IS to TO-BE EAs. The main component of extended visual and ontological modeling is an extended enterprise metamodel that integrates descriptive and prescriptive concepts. Our ongoing model building effort with a large real world case study suggests that visual modeling editor simplifies and streamlines modeling process and ontological models enable expressing requisite analyses with ease.

The paper is arranged as follows. In section 2, we describe the extended visual and ontological modeling support. Section 3 explains how visual and ontological models are used together in charting a course of action from AS-IS architecture to desired TO-BE architecture. Section 4 recounts related work and concludes the paper.

2 Visual and Ontology Modeling for Extended Enterprise Models

Figure 1 shows the steps from modeling intentions and motivations of an enterprise (henceforth called *IM modeling*) in its response to a change till the operationalization of chosen strategic alternative. Figure 1 also shows specific visual models and analyses in each step.

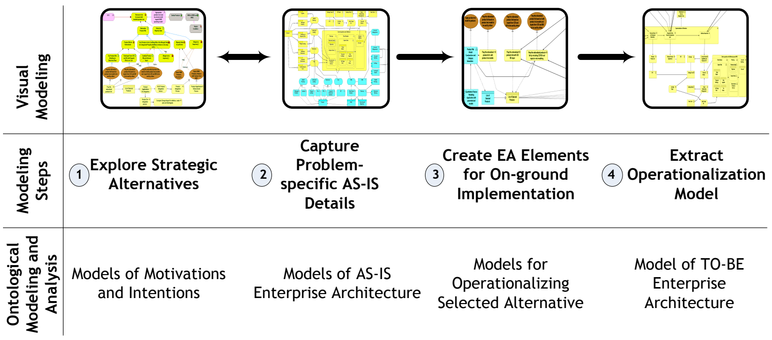


Fig. 1: Steps in Enterprise's Response to Change

In the following, we first describe the extended enterprise metamodel and how it is used as the basis of visual and ontological modeling. We then describe the steps in Figure 1 briefly.

2.1 Extended Enterprise Metamodel

The mapping between core metamodels of ArchiMate and i* enabled treating intentional modeling as a problem solving technique for enterprises as described in [1]. As we modeled a number of in-home enterprise scenarios, we found that a distinction was needed between the concepts of motivation or driver and goal. Motivation is something that would lead an enterprise to consider certain goals. In practical enterprise modeling, concepts of drivers, stakeholders, and assessment make more sense to domain experts than notions of goals and soft goals. Figure 2 shows the extended enterprise metamodel which integrates ArchiMate's core metamodel with IM modeling concepts.

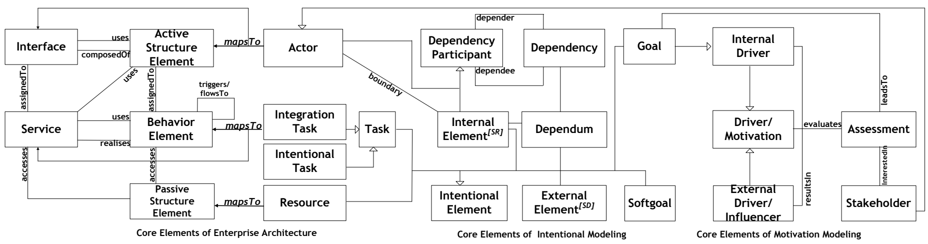


Fig. 2: Extended Enterprise Metamodel with EA and Intentional [1] and Motivational Concepts

While Figure 2 shows this mapping at conceptual level, various cardinalities are made explicit in the underlying ontology as described later in Section 2.3. This mapping enabled us to specify that *enterprise active structure entities are motivated by internal and/or external drivers to use or create passive structure entities while performing behavior (entities) as means to ends that are goal(s) or soft goal(s)*. This kind of articulation of enterprise's problem context is streamlined using visual modeling support as we explain next.

2.2 Visual Modeling Support for EA and IM Models

As described earlier, we used open source tools Archi and OpenOME for enterprise and intentional modeling respectively. We chose Archi for extended enterprise modeling as it already supports modeling the business, application, and infrastructure layers of ArchiMate [2] and we would have to add only the IM concepts to it. With the core metamodel of Archi extended to metamodel in Figure 2, it becomes possible to visually model both EA and IM concepts together.

In Figure 3, 1 shows models of products and services rationalization problem in a case study of merger and acquisition (M&A) of two large wealth management banks which we introduced in [5]. We are modeling this case study in terms of models specific to problems described in [5] consisting of over a 1500 entities and more than 2500 relations so far.

The extended Archi elements based on the extended enterprise metamodel in Figure 2 are shown by 2 in Figure 3. Archi is itself based on Eclipse Modeling Framework (EMF) and makes addition of elements and relations easier on top of business, application, and infrastructure metamodels of EA that it already provides [2].

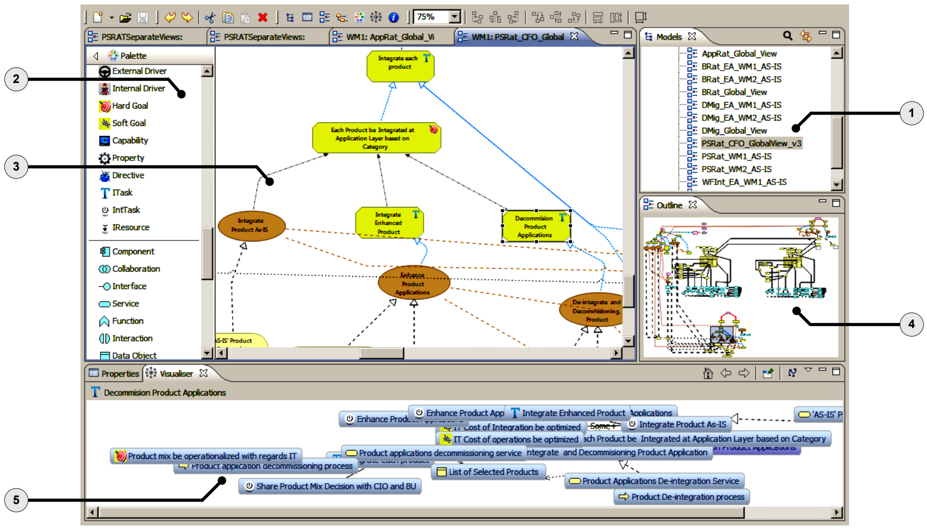


Fig. 3: Extended Archi for EA and IM Modeling

The process of adding elements to base metamodels is straightforward in which EMF-based classes have to be added followed by code generation etc. A relation is added in Archi, by internally assigning a specific character that recognizes a relation and then the source types mention the character to enable using this relation with the allowed target element. By defining all the elements required by EA and IM models and sets of allowed relations between various elements, EA and IM models can be drawn as shown by 3 in Figure 3. The models panel in Figure 3 enlists AS-IS EA models as well as IM models of these problems we refer to as global views. 4 shows such a global view of products and services rationalization problem. Current version of Archi supports visualization of a selected model element in terms of all other elements that it is related to as shown by 5 in Figure 3. An example of EA and IM models drawn in the context of aforementioned case study is shown in Figure 4.

We need to keep the details of the case study outside the scope of the paper due to space restriction. Suffice it to say that the extended visual modeling elements proved to be considerably useful when modeling problem-specific aspects of the case study with domain experts. Both the implicit reasoning that the domain experts used and the existing EA elements could be modeled in the same view. This enabled clear flow of domain experts' understanding that is translated to models conforming to the extended metamodel.

2.3 Ontology Modeling Support for EA and IM Models

We presented an ontological representation that captures ArchiMate's core metamodel as well as layer specific metamodels in [4]. This representation was versatile enough for conducting change impact and landscape mapping analyses. We extended the EA ontology presented in [4], with IM concepts as shown in Figure 5. For EA ontology modeling and analyses, reader is requested to refer to [4]. We only explain ontological modeling of IM concepts here. Archi enables export to CSV files which retain EA element type and name of both source and target nodes along with relation and documentation if any. While custom exporters can be easily written, the default export option was quite sufficient for our purposes. The exported model is easily read into ontology model by first constructing the dictionary leveraging the type information of instances and then constructing the data in terms of relations.

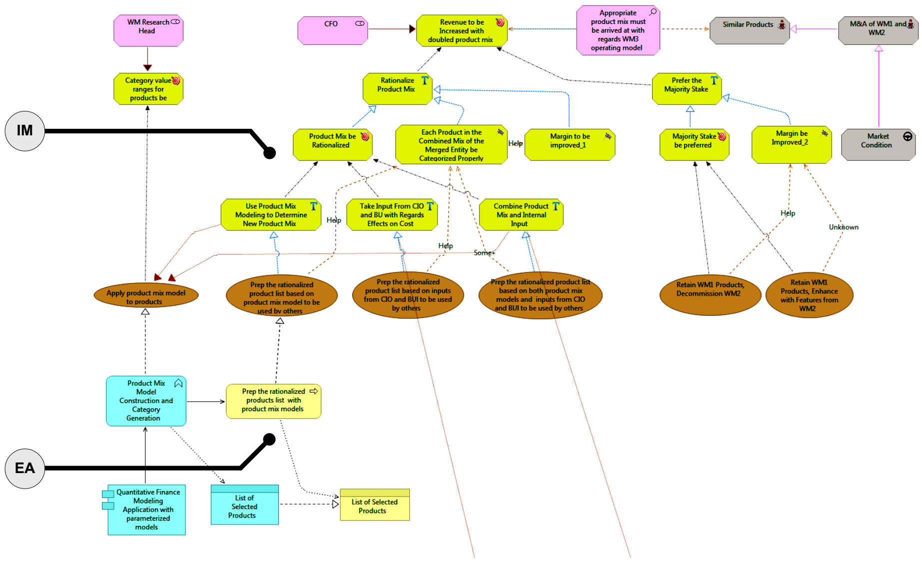


Fig. 4: *IM* and *EA* Model of Chief Financial Officer *Actor* in Products and Services Rationalization Problem

Ontological Representation of IM Concepts Intentional modeling concepts are captured under IntentionalEntity class as shown in Figure 5. The metamodel in Figure 2 distinguishes between elements internal to an actor (comprising strategic rationale model) and external to actors (strategic dependency model) [6]. We found that all of the relations in intentional model namely, means-end (MELink), task decomposition (TDLink), contribution (CTLink), and strategic dependency (SDLink) relations, benefit from being represented as *reified* relations [7]. For instance, a contribution link indicates not only which element contributes to a soft goal but also what that contribution is.

We have chosen those motivational concepts and relations from [8, 9], which we think have practical relevance as shown in Figure 2. These are captured under MotivationalEntity ontology class. Drivers, both within an enterprise and from the enterprise's environment, influence rest of the IM elements. Generally, a stakeholder becomes interested in assessment of a driver and it is this assessment that leads to formulation of a goal. From thereon, intentional modeling begins in terms of actor who is responsible for achieving the goal and actions that need to be taken by that actor, in some cases depending on other actors.

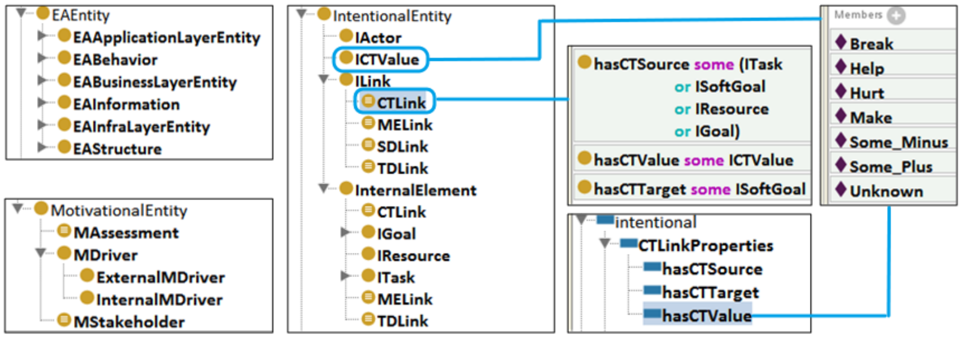


Fig. 5: Ontological Representation of Extended Metamodel in Figure2

3 Using Visual and Ontology Modeling

In the following, we describe in brief how visual and ontology modeling is used in a step by step manner as previously indicated in Figure 1.

Step 1- Exploring Strategic Alternatives In our approach, IM modeling is carried out first to gain clarity of the problem context. The AS-IS architecture of the enterprise with elements pertaining only to the problem situation at hand could be then easily modeled.

Upon completion of the IM modeling activity for a given problem, leaf tasks can be assigned satisfaction levels. We have implemented the bottom-up label propagation algorithm in [10] to compute satisfaction level of the root goal. The ontological representation easily enables implementing the label propagation as well as computation of specific routines. A SPARQL[1] query to get all the immediate tasks that are means to a specific IHardGoal instance is shown in Listing 1.1.

Listing 1.1: Traversing Reified Means-Ends Links

```
"SELECT ?task " + "{ " +
"?s rdf:type :IHardGoal . " + "?s :name \"" +
<IHardGoal> + "\" . " +
"?meLink :hasMEGoal ?s ." +
"?meLink :hasMETask ?task ." +
" }";
```

Step 2- Capturing Problem-specific AS-IS Details We showed how EA ontology for given enterprise model can be leveraged to query the model in [4]. Using what-if analyses of IM models and queries on enterprise models and going back and forth between these models, both models can be refined such that they capture the reality to the satisfaction of domain experts.

Step 3- Creating EA Elements for On-ground Operationalization In the next step, leaf tasks in IM models are related to EA elements that will operationalize them from TO-BE EA perspective. IM alternatives can be combined in various ways based on whether they satisfy the root goal. In order to preserve which elements were added

[1] http://www.w3.org/TR/rdf-sparql-query/

for specific leaf tasks, we tag them in the ontological representation using an object property called isOperationalizationElementOf. This achieves two purposes- first, the EA elements that are added anew in contrast to AS-IS elements are identified; and second, which new elements constitute the operationalization of a specific leaf task is made explicit.

Step 4- Extracting Desired TO-BE Operationalization Model The specific set of elements that would comprise an operationalization model of a specific strategy is computed by creating an extended Archi viewpoint only of the operationalization models of all alternatives and exporting it as the sole model to import into ontological representation. Using this as the base model, the set of operationalization elements of a specific strategy can be easily separated out using a SPARQL query over the tagging introduced in Step 3.

4 Related Work and Conclusion

Various approaches have suggested combined treatment of enterprise and IM concepts [8, 11, 12], but with either the visual modeling support or the programmatic means of analyzing models missing, they remain without use in practice for large scale enterprise what-if analyses. Similarly, visual modeling support remains one sided- either IM modeling support is at an early stage as in [13] or EA based modeling is lacking as in [14].

Our ongoing experience in modeling the M&A case study suggests that visual modeling support for extended enterprise models streamlines the case modeling activity while ontology modeling support enables implementing requisite analyses. Visual models are imported into ontology and therefore models and analysis results remain in sync. While visual modeling support explained in the paper is working as desired, we suspect that when many modelers are simultaneously modeling various problem-specific IM and enterprise models in multiple interactions with domain experts, a more robust distributed enterprise and IM modeling environment will be necessary. We are actively pursuing the possibility of implementing such an environment using our proprietary reflexive metamodeling framework which we have used successfully in several business engagements [15].

References

1. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Intentional Modeling for Problem Solving in Enterprise Architecture. In Hammoudi, S., Maciaszek, L.A., Cordeiro, J., Dietz, J.L.G., eds.: ICEIS (3), SciTePress (2013) 267–274
2. Institute, E.C.: Archi Developer Resources. `http://archi.cetis.ac.uk/developer.html` Accessed: 16th March, 2014.
3. Horkoff, J., Yu, Y., Yu, E.S.K.: OpenOME: An Open-source Goal and Agent-Oriented Model Drawing and Analysis Tool. [16] 154–156
4. Sunkle, S., Kulkarni, V., Roychoudhury, S.: Analyzing Enterprise Models Using Enterprise Architecture-Based Ontology. In Moreira, A., Schätz, B., Gray, J., Vallecillo, A., Clarke, P.J., eds.: MoDELS. Volume 8107 of Lecture Notes in Computer Science., Springer (2013) 622–638

5. Sunkle, S., Kulkarni, V., Rathod, H.: (Multi-) Modeling Enterprises for Better Decisions. In Combemale, B., DeAntoni, J., France, R.B., Barn, B., Clark, T., Frank, U., Kulkarni, V., Turk, D., eds.: GEMOC+AMINO@MoDELS. Volume 1102 of CEUR Workshop Proceedings., CEUR-WS.org (2013) 69–78
6. Ayala, C.P., Cares, C., Carvallo, J.P., Grau, G., Haya, M., Salazar, G., Franch, X., Mayol, E., Quer, C.: A Comparative Analysis of i*-Based Agent-Oriented Modeling Languages. In Chu, W.C., Juzgado, N.J., Wong, W.E., eds.: SEKE. (2005) 43–50
7. Group, W.W.: Defining N-ary Relations on the Semantic Web. `http://www.w3.org/TR/swbp-n-aryRelations/` W3C Working Group Note 12 April 2006 Accessed: 16th March, 2014.
8. Quartel, D., Engelsman, W., Jonkers, H.: ArchiMate ® Extension for Modeling and Managing Motivation, Principles and Requirements in TOGAFT™ (October 2010)
9. OMG: Business Motivation Model - Version 1.1. (May 2010)
10. Horkoff, J., Yu, E.S.K.: Evaluating Goal Achievement in Enterprise Modeling - An Interactive Procedure and Experiences. In Persson, A., Stirna, J., eds.: PoEM. Volume 39 of Lecture Notes in Business Information Processing., Springer (2009) 145–160
11. Yu, E.S.K., Strohmaier, M., Deng, X.: Exploring Intentional Modeling and Analysis for Enterprise Architecture. In: Tenth IEEE International Enterprise Distributed Object Computing Conference (EDOC) Workshops. (2006) 32
12. Engelsman, W., Wieringa, R.: Goal-Oriented Requirements Engineering and Enterprise Architecture: Two Case Studies and Some Lessons Learned. In Regnell, B., Damian, D.E., eds.: REFSQ. Volume 7195 of Lecture Notes in Computer Science., Springer (2012) 306–320
13. Frank, U.: Multi-Perspective Enterprise Modelling as a Foundation of Method Engineering and Self-Referential Enterprise Systems. In Hammoudi, S., Maciaszek, L.A., Cordeiro, J., Dietz, J.L.G., eds.: ICEIS (1), SciTePress (2013) IS–9
14. Morandini, M., Nguyen, D.C., Penserini, L., Perini, A., Susi, A.: Tropos Modeling, Code Generation and Testing with the Taom4E Tool, booktitle = iStar. [16] 172–174
15. Kulkarni, V., Reddy, S., Rajbhoj, A.: Scaling Up Model Driven Engineering - Experience and Lessons Learnt. In Petriu, D.C., Rouquette, N., Haugen, Ø., eds.: MoDELS (2). Volume 6395 of Lecture Notes in Computer Science., Springer (2010) 331–345
16. de Castro, J.B., Franch, X., Mylopoulos, J., Yu, E.S.K., eds.: Proceedings of the 5th International *i** Workshop 2011, Trento, Italy, August 28-29, 2011. In de Castro, J.B., Franch, X., Mylopoulos, J., Yu, E.S.K., eds.: iStar. Volume 766 of CEUR Workshop Proceedings., CEUR-WS.org (2011)

Eclipse SoaML: a Tool for Engineering Service Oriented Applications

Andrea Delgado, Laura González

Instituto de Computación, Facultad de Ingeniería, Universidad de la República

Julio Herrera y Reissig 565,11300, Montevideo, Uruguay
{adelgado, lauragon}@fing.edu.uy

Abstract. Service-Oriented applications focus on the definition of self-contained independent pieces of software called services providing specific functionality, being nowadays one of the most used implementations to support business process execution. Although languages, technologies and platforms supporting the service-oriented paradigm have evolved within the last decades, it is not the case of service modeling support. Service modeling is a key aspect for the design and specification of services and for automating the different stages following the model driven development vision. SoaML is a standard for service modeling which defines an UML profile and a metamodel extending the UML metamodel, defining explicitly concepts and elements for services specification. The Eclipse SoaML tool provides support for service modeling with UML using the SoaML standard, allowing importing and exporting models in XMI format to interoperate with other tools. From SoaML models the tool also generates the code needed to implement services in Java EE and Web Services.

Keywords: Service Oriented Architectures (SOA), service modeling, service development, standards, SoaML, JEE, WS.

1 Introduction

Service Oriented Computing (SOC) [1] promotes the design of applications based on services that are reusable software components by means of which consumers and providers interact in a decoupled way. A Service Oriented Architecture (SOA) [2][3] is an architectural style that supports service orientation providing the means for designing and specifying services. Services are usually defined as self-contained independent pieces of software providing specific functionality, being nowadays one of the most used implementations to support the automatic execution of business processes. Business Process Management [4][5] provides the means for modeling, executing and assessing business processes and their implementations in organizations.

While languages, technologies and platforms support for the implementation and execution of services is an area that in recent years has evolved considerably, the design and modeling of services is still immature. Service modeling is essential among other things, for automating various stages of software development using

Model Driven Development (MDD) [6][7], such as code generation. Service models can be enhanced with platform specific information to obtain the corresponding code, providing traceability from requirements to code which eases software maintenance. The Service Oriented Architecture Modeling Language (SoaML) [8] standard is based on UML and is one step in the direction of providing support for service modeling by defining specific concepts and stereotypes for modeling services. Currently there is a lack of support for the SoaML standard since there are no more than a few implementations of it which are mostly commercial tools[1].

The Eclipse SoaML tool we present here is composed of two Eclipse plug-ins which provide support for engineering service-oriented applications: the first one is a SoaML editor which implements the SoaML standard allowing service modeling in SoaML and interchange of service models in XMI format; the second one is a code generator for implementing services with Java EE and Web Services which takes as input a SoaML service model and generates the corresponding code. Both Eclipse plug-ins were developed as part of a broader research work regarding the continuous improvement of business processes implemented by services with a model-driven focus, reflected in the definition of the framework MINERVA [9][10][11][12][13].

The rest of the article is organized as follows: in section 2 we describe the SoaML standard, in section 3 we present the Eclipse SoaML Tool we have developed to support SoaML modeling and code generation, in section 4 we discuss related work, and finally in section 5 we present some conclusions and future work.

2 SoaML standard

The SoaML standard for service modeling has been recently released in its version 1.0.1 (May, 2012) and its first beta release was only five years ago (April, 2009). It defines the concepts and corresponding elements and stereotypes needed for modeling service-oriented applications. The most important one is a *Service,* which consist of a value offering according to one or more defined *Capabilities*. It provides an *Interface* with *Operations*, with input and output *Parameters* and associated *Types*, and a *ServiceContract* which defines the roles and interfaces involved in the execution of the service. A *ServicesArchitecture* is a UML *Collaboration* which shows the *Participants* of the network or services community, the *ServiceContracts* of the *Services* for interaction between each other, and the roles each one plays within each service.

Participants may be software components, organizations, or systems that provide and use services; services are provided by means of *Service Ports* and requested by means of *Request Ports* which are specializations of UML *Ports*. Services can be modeled with a *ServiceInterface* or simple UML *Interface*s which define the elements needed to interact with the service. The *ServiceContract* defines the interfaces, roles and choreography that participants agree to use to interact within each other. A *ServiceChannel* models the communication between consumers and providers, and the *MessagesType* allows specifying the information exchanged within the operations.

---

1 SoaML implementations, http://www.omgwiki.org/SoaML/doku.php

3 Eclipse SoaML Tool

Both Eclipse plug-ins of the Eclipse SoaML Tool were developed as part of a broader research line as mentioned before, the MINERVA framework. The first one focuses on service modeling and the second one on generating code from these SoaML models. Other tools and proposals developed within this wider work to support our vision based on business processes implemented by services are not shown here.

3.1 SoaML Tool Architecture

The Eclipse environment allows the integration of several tools by means of developing plug-ins which can be directly downloaded and installed in the IDE. Along with the flexibility this provides for integrating specific tools, Eclipse is one of the most used IDEs within the software development community, and provides several options for UML modeling and Web Services generation. It provides extension points to which new plug-ins can be "hooked" reusing existing environment implementations. In Fig. 1 the Eclipse Architecture is shown along with the integration with the SoaML plug-ins composing the SoaML Tool.

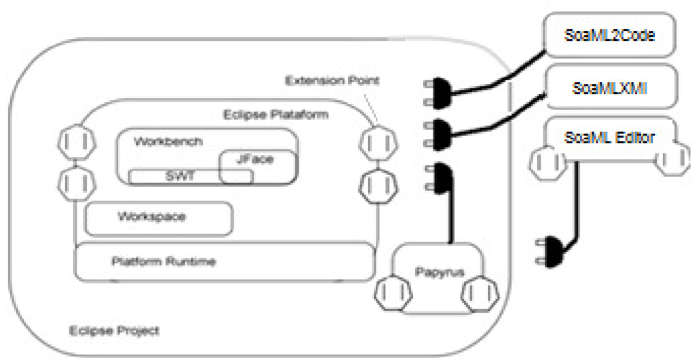


Fig. 1. Eclipse Architecture with extension points and integration of the SoaML Tool

3.2 SoaML editor plug-in

The Eclipse SoaML[2] plug-in for service modeling is developed above the existing Eclipse Papyrus[3] UML 2 modeler, as shown in Fig. 1. Papyrus provides the base UML metamodel and implementation of palette elements, to which we added the SoaML specific extensions and elements. Papyrus is composed of several plug-ins (.jars) implementing the different UML elements and diagrams provided by the editor. For the solution we provide we have reused the code corresponding to the class and composed structures of Papyrus, creating new plug-ins (.jars) for each type of SoaML diagram to be added to the base Papyrus tool.

[2] http://www.fing.edu.uy/inco/grupos/coal/en/field.php/Proyectos/EclipseSoaML
[3] Papyrus UML, http://www.papyrusuml.org/

SoaML editor functionalities.

The SoaML editor plug-in provides seven diagrams to support the elements defined in the SoaML standard: ServicesArchitecture, Participants (Class and Component), Interfaces, MessagesTypes, ServiceContracts and Capabilities. Fig. 2 shows the options for creating SoaML diagrams where the top ones correspond to UML diagrams from Papyrus and the bottom ones (a) correspond to the SoaML diagrams we have added. In the Model Explorer view from the Papyrus perspective which is shown at the left (b) in Fig. 2, you can see the tree showing the structure and elements of the SoaML model, which are visualized on the diagram layouts to the right (c).

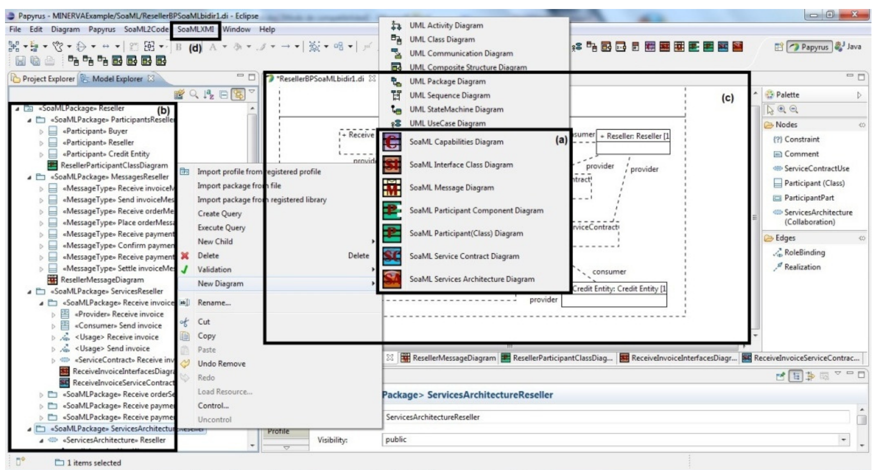


Fig. 2. Screenshot of the Eclipse SoaML editor showing diagrams and a SoaML model

The Eclipse SoaML plug-in also allows importing and exporting SoaML models in XMI format, to interoperate with other SoaML tools, an option that is provided in the Eclipse menu as shown in Fig. 2 (d). This means that we can import and visualize a service model specified in SoaML with another tool, and we can export a SoaML service model in XMI to be visualized in another tool. Some tools add specific tags that are not included in the SoaML standard (such as MagicDraw[4]) so in these cases the interoperability is affected and the XMI file must be manually "rearranged" to be imported. Interoperability was successfully tested with Modelio early versions[5].

Fig. 3 (a) shows as an example how the SoaML ServicesArchitecture model corresponding to the Reseller business process adapted from [4] is visualized in the SoaML editor. Fig. 3 also shows examples of other SoaML diagrams: (b) Participants showing ports in which services are provided and required, (c) service Interface showing the provider and consumer interfaces with operations, parameters and types for the service "Receive Invoice", and (d) ServiceContract showing the defined roles.

4 http://www.nomagic.com/products/magicdraw-addons/cameo-soa.html
5 http://www.modeliosoft.com/en/modelio-store/type/free.html

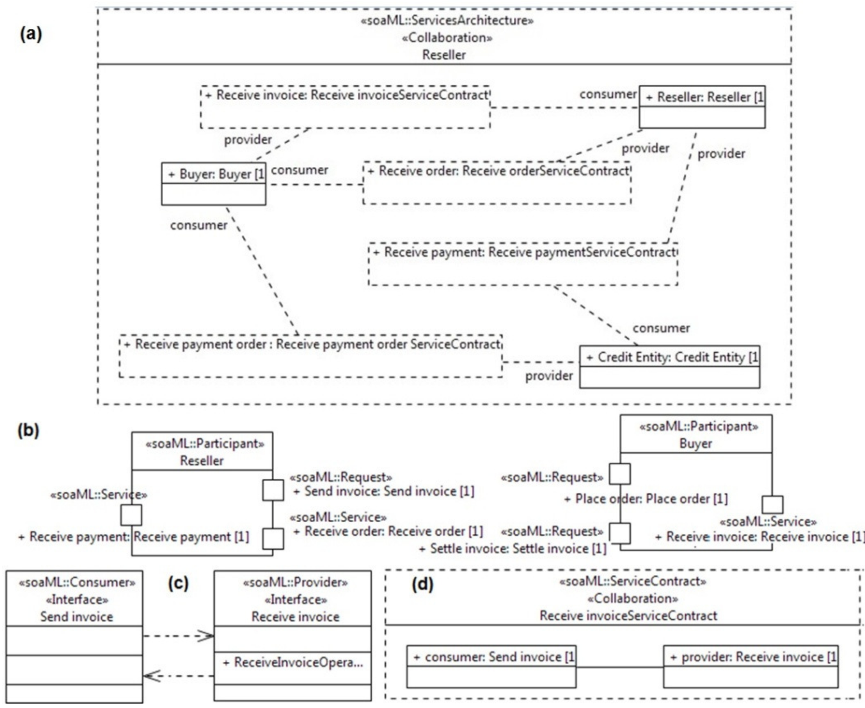


Fig. 3. SoaML diagrams: (a) ServicesArchitecture, (b)Participants, (c) Interfaces and (d) ServiceContract

3.3 SoaML2Code plug-in

The Eclipse SoaML2Code[6] plug-in generates specific java projects from SoaML service models, both for service providers and consumers. It requires as input an XMI file containing a SoaML service model, which can be modeled in the same Eclipse SoaML Tool using the SoaML editor plug-in presented above, or another SoaML tool or generated automatically from a business process model in MINERVA framework. Fig. 4 presents the three components that were designed and implemented in order to generate code from SoaML models. It also shows the interaction with the SoaML plug-in or any other tool exporting SoaML models in the XMI or UML formats.

The “SoaML models to Java objects converter” is in charge of parsing the input file with the SoaML model, in order to generate Java objects which represent it. The input file can be an XMI file (version 2.0 or 2.1) or an UML file. The “SoaML models Processor” is in charge of processing the previously generated Java objects and applying to them a set of rules and conditions for invoking the code generators. Currently, the plug-in supports JEE and WS generators, but new generators may be incorporated. The user has different generation options, for example, the type of project to generate (i.e. client or server), the target application server (JBoss AS or Apache

[6] http://www.fing.edu.uy/inco/grupos/coal/en/field.php/Proyectos/EclipseSoaML2Code

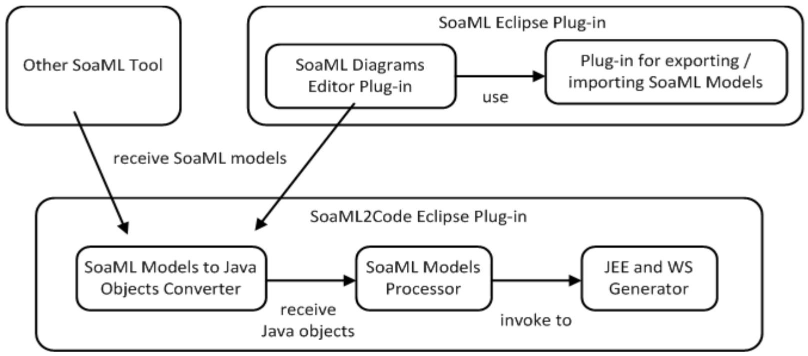


Fig. 4. Components for generating code from SoaML models

Tomcat) and, in case of generating a server project, the type of application which will expose the services (i.e. a JEE application or a Web application). The "JEE and WS Generator" is in charge of generating, based on the SoaML model, the JEE or WS code by invoking the existing generators. Fig. 5 shows the SoaML2Code plug-in, in particular, it shows (a) the available options to generate Java objects from a SoaML model as described above, and on the left side (b) the XMI and Papyrus files corresponding to the SoaML model and layout.

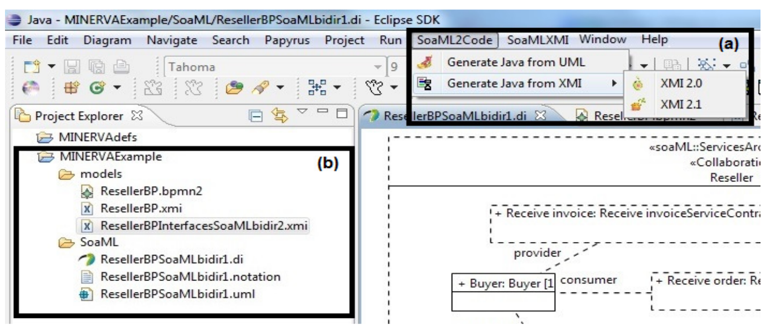


Fig. 5. Eclipse SoaML Tool with the plug-in for code generation from SoaML Models

SoaML2Code functionalities.

From the SoaML diagrams as presented in Fig. 3, the SoaML2Code plugin can be used to generate JEE or WS code by selecting the menu option "Generate Java from JEE" or "Generate Java from UML", as shown in Fig. 5. After selecting one, the generation options (e.g. target application server) are presented in a dialog for the user to choose. For the purpose of this example, we assume that the following options were selected: JBoss AS as the target application server and a Web application to expose the services. Fig. 6 (a), shows on the left a Java project generated for each participant of the SoaML model. Also, for each Port whose interface has the provider role (c.f. Fig. 3), the Java interfaces and classes are generated in order to expose the service as a Web Service. The right side of Fig. 6 (a) shows how the Java code of these interfaces and classes is enriched with several annotations (@WebService, @WebMethod, etc) specified in the JSR 181 (Web Services Metadata Annotations). The SoaML2Code plug-in also generates the WSDL declarations for each of the services. Fig. 6 (b) shows an extract of the WSDL file for the ReceivePayment service.

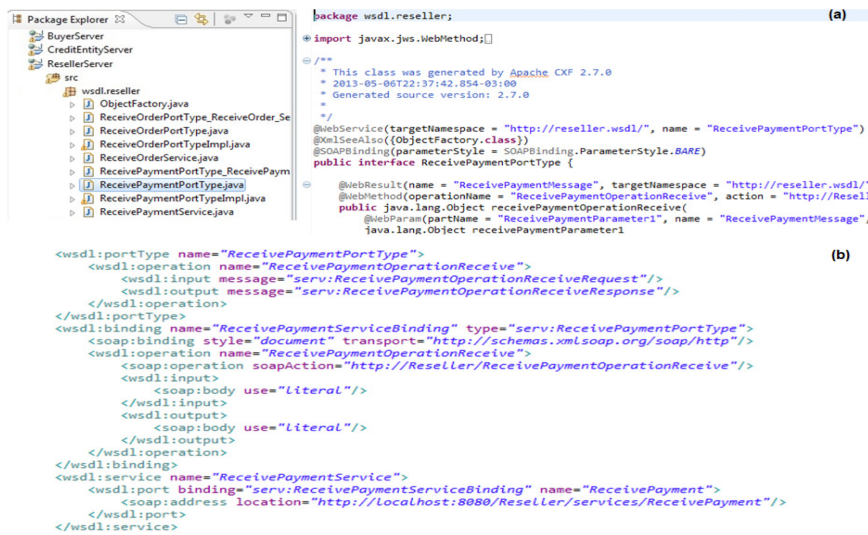


Fig. 6. Generated Eclipse projects and Java Code (a) and extract of a WSDL file (b)

4 Related work

To the best of our knowledge there are few implementations of the SoaML standard, mostly commercial ones such as: IBM Rational Architect[7], Visual Paradigm[8], Sparx Systems' Architect[9] and Magic Draw[10]. The last one provides a SoaML editor which can be used together with the open source ModelPro[11] engine to generate code, although ModelPro seems to be a discontinued project. Modelio[12] provides an open source SoaML editor but to be used within their commercial solutions. Our approach is an open source one integrated in the Eclipse environment, which provides easy installation and use within an IDE widely used in the software community, along with existing plug-ins such as Papyrus and WS and Java generators. It was validated by means of case studies based on real service models to support business processes.

---

[7] http://www.ibm.com/developerworks/downloads/r/architect/index.html
[8] http://www.visual-paradigm.com/product/vpuml/features/soamlmodeling.jsp
[9] http://www.sparxsystems.com.au/products/ea/index.html
[10] http://www.nomagic.com/products/magicdraw-addons/cameo-soa.html
[11] http://portal.modeldriven.org/project/ModelPro
[12] http://www.modeliosoft.com/en/technologies/soa.html

5 Conclusions and future work

We have presented the Eclipse SoaML Tool which provides support for service modeling with SoaML, XMI import and export of SoaML models and generation of code from SoaML models for JEE and WS. We believe it is a useful tool to provide support for service modeling, which is a key aspect in the development of robust SOA applications and for supporting business process execution. Modeling SOAs and automatically generating code from these models, allow registering traceability from business concepts to services and from services to code, easing software maintenance, reuse of services and interchangeability of implementations. We are now working on extending SoaML models with QoS characteristics and generating the related code.

Acknowledgements: This work was partially funded by the Agencia Nacional de Investigación e Innovación (ANII), Uruguay, project ANII-FCE-2011-7341. We would like to thank students Andrés Pastorini, Sofía Larroca, Natalia Maurizio, Victor Fuica and Alejandro Goss for their work within the implementation of the tools.

References

1. Papazoglou, M.; Traverso, P.; Dustdar, S.; Leymann, F.: Service-Oriented Computing: State of the Art and Research Challenge, IEEE Computer Society, (2007)
2. Krafzig, D. Banke, K. Slama, D., Enterprise SOA: Best Practices, Prentice Hall, (2005)
3. Erl, T.,SOA: Concepts, Technology, and Design,Prentice Hall, (2005)
4. Weske, M., BPM Concepts, Languages, Architectures, Springer, (2007)
5. van der Aalst, W.M.P., ter Hofstede, A., Weske, M., Business Process Management: A Survey, In: International Conference on Business Process Management, (2003)
6. Mellor, S., Clark, A., Futagami, T., Model Driven Development - Guest editors introduction, IEEE Computer Society, September/October (2003)
7. Model Driven Architecture (MDA), OMG, http://www.omg.org/mda/specs.htm, (2003)
8. Soa Modeling Language (SoaML), OMG, http://www.omg.org/spec/SoaML/, (2009)
9. Delgado A., Ruiz F., García-Rodríguez de Guzmán I., Piattini M:, MINERVA: Model drIveN and sErvice oRiented framework for the continuous BP improVement & relAted tools, In: 5th Int. Work. Engineering SO Applications (WESOA'09), (2009)
10. Delgado, A., Ruiz, F., García-Rodríguez de Guzmán, I., Piattini, M.; "Business Process Service Oriented Methodology (BPSOM) with Service generation in SoaML". In 23rd Int. Conf. on Advanced IS Engineering (CAiSE), (2011)
11. Delgado, A., Ruiz, F., García-Rodríguez de Guzmán, I., Piattini, M., "Model Transformations for Business-IT Alignment: From Collaborative BPs to SoaML Service Model", In: 27th ACM Symposium on Applied Computing (SAC), (2012)
12. Delgado; A., Weber, B., Ruiz; F., García; F., Piattini, M., "An integrated approach based on execution measures for the continuous improvement of BPs realized by services", Information and Software Technology Journal, Elsevier, Vol. 56, 2, p. 134–162, (2013-14)
13. Delgado, A, González,L., "Automatic generation of SOAs for Business Process execution: A vision based on models, Latin American Computing Conference (CLEI), (2013)