Publishing and Discovery of intentional services: Goal-driven approach

Kadan Aljoumaa

Centre de Recherche en Informatique, Université Paris 1 - Panthéon Sorbonne,
90, rue de Tolbiac
75013 Paris, France
kadan.aljouma@malix.univ-paris1.fr

Supervised by Professor Carine Souveyet and Associate professor Saïd Assar

Abstract. With the increasing growth in popularity of Web services, discovering relevant Web services becomes a significant challenge. The introduction of intentional services is necessary to bridge the gap between low level, technical software-service descriptions and high level, strategic expressions of business needs for services. Current Web Services technology based on UDDI and WSDL does not make use of this "intention" and therefore fails to address the problem of matching between capabilities of services and business user needs. In this research work, we propose publishing and discovery of services based on intentions they satisfy by extending the actual registry to build the intentional one. We also focus on the part of discovery of intentional services concerning the formulation of the user's requests. Our contribution consists of proposing a goal meta-model for user's request using ontologies, the descriptor to be published in the registry and the matching algorithms between services' intentions and users' goals using similarity metrics.

Keywords: intentional service, goal, intentional service publishing, intentional service discovery, Ontology.

1 Introduction

Within a decade, Web services and SOA became a viable technical solution for the development of information systems. They provide flexibility in maintenance and evolution of systems, and ensure a high degree of interoperability between heterogeneous systems [14] [15]. A major problem in the use of SOA is the discovery of appropriate services that meet business needs [9]. This difficulty will increase with the number and variety of web services available online. The technological tools currently available (such as WSDL for Web Service Description Language and UDDI for Universal Description Discovery and Integration) are semantically poor and conceptually far from the concerns of the user. As many writers have observed, there is a "conceptual mismatch" between the service provider side located on operational level, and the user's needs whose are expressed in business terms.

The work presented in this thesis is built on earlier work in which the model $\mathcal{ISM}$ (Intentional Services Model) has been proposed for modeling and describing services in business terms [9], [10], [19]. $\mathcal{ISM}$ shares with other approaches the need to describe service to ease their retrieval but departs from their *function driven* perspective to propose an *intention drive* service description. As a consequence, $\mathcal{ISM}$ service descriptions will bring out the business intention that the service allows to fulfill with pre and post conditions instead of defining the signatures of operations that can be invoked on class objects. We believe that this approach contributes to resolve the current mismatch of languages between low level services descriptions such as WSDL statements and business perceived services.

Therefore, this work is part of the platform iSOA and specifies the capacity to search services in the registry using goal driven approach. It is related to the problem of publishing, querying and discovering of intentional services based on user goal and on the service intention.

We use this framework and this methodology to specify and implement the registry of the iSOA platform. Specifically, it will lead us to choose the most effective techniques for intentional services discovery and suggest matching algorithms to select the services that best meet user needs.

This article is organized as follows: first we present the research question concerning the discovery of services based on the goal they satisfy. Then, we introduce the proposed solution to this problem. The idea consists of proposing a goal meta-model used for formulating user query, the ontologies needed in the discovery and finally, the intentional descriptor required for intentional services publishing will be defined and implemented in an extended registry.

Before we conclude our work, we present the related work in this domain and how we compare other approaches to our solution.

2 Research question

The problem addressed in this thesis deals with the publishing and discovery of intentional services. Users formulate their needs in natural language in which they express a query with one goal to be achieved (Fig. 1).

The reply of natural language queries raises two major problems:

- Several intentional services candidate can match this query with variable degrees of similarity. For example, a query such as "*gather information*" may correspond to the intentional service "*integrate data*" with a degree of similarity 85% and another intentional service "*reassemble files*" with a degree of similarity 45%.
- The answer to a query could be an individual intentional service, as it could be in some cases a group of several intentional services that the user may compose to meet its needs. For example, "*accept delivery of goods*" may correspond to the intentional services "*acquire products*", "*enter the product in stock*" and "*enter in stock the delivered goods"*.

The solution of these problems requires three steps: (i) defining a service descriptor that allows the intentional publication in a registry, (ii) developing a query language

that helps to discover intentional services published in the registry and (iii) specifying an approach to perform intentional service discovery based on similarities calculation.

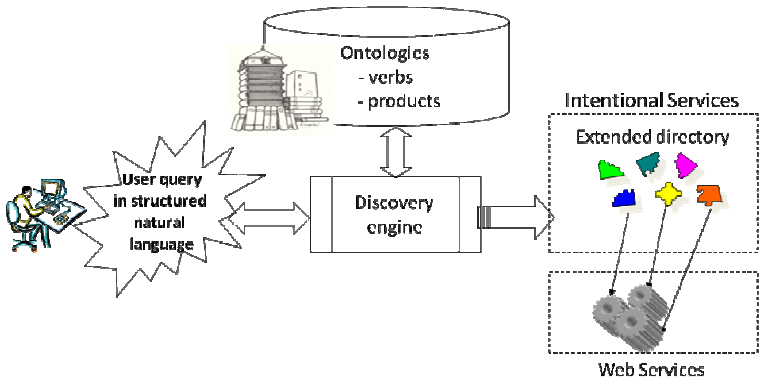

Fig. 1. *Intentional services Querying*.

The first key issue of this thesis focuses on specifying the intentional descriptor to be published in the registry and on the publication of aggregated intentional services. Additionally, it seeks to couple intentional service with software services executable by market standards. We rely on the concept of intentional atomic service which enriches the operational web service[1] with the concept of intention.

2.1 Meta-model of intentional services

The publishing of intentional services is the first step toward building the intentional registry. For this reason, we use the intentional model ISM [9] to present the services in the registry where the concept of goal is originally used in this model. An *intentional service* is a service captured at the business level, in business comprehensible terms and described in an intentional perspective, i.e. focusing on the intention it allows to achieve rather than on the functionality it performs. The model defines each intentional service as building brick in the application by associating it with the situational knowledge in the interface. Each intentional service fits a particular situation in order to achieve a particular intention.

The model of intentional service of [9, 19], takes the form of a composition of services based on graphs, AND/OR tree of goals. The composition of services driven by goals introduces a composition on several levels: for the highest service level, which may be strategic in nature, is broken down itself into sub/intentional services, may require a new de/composition to achieve the intentional services (Fig. 2).

[1] Web services are one possible technological solution for implementing intentional service. But this notion of intentional service is not limited to a specific technology.

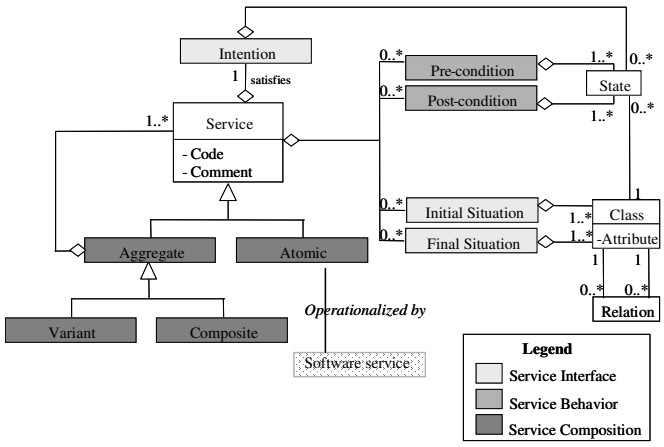

Fig. 2. *Intentional Service Model [9, 19].*

2.2 User query

We assume that the user needs are driven by goals, and are expressed using a language easily understandable by non-expert domain users, which is different from service models definition languages that require technical knowledge of the area.
In this thesis, we focus on user needs expressed in structured natural language. We propose an ontology based solution for the interpretation of these needs. The implementation of the ontology requires the prior definition of a goal model for expressing queries. This model will allow the exploitation of ontology to find similarities between the components of the user query and the attributes that make up the targeted intentional services published in the registry. In addition, we could reformulate user needs in order to enlarge the searching possibilities.

2.3 Discovery of intentional services

The third issue of interest is service discovery in the registry. The methodology proposed in [9] has no mechanisms for finding services at intentional level in the registry. A process proposal that aims at locating intentional services can contribute to enriching the methodology. Finding intentional service consists in establishing the matching between the goal (what a user of the registry is seeking to achieve) and intention displayed by the service (what the service guarantees to meet).

In our work, we propose mechanisms for finding intentional services available in the registry. The finding process can integrate several aspects including:

- The intentional services retrieval, i.e. select services in the registry based on the characteristics of the descriptor.
- Metrics of similarity: We need to measure how two intentions are semantically similar even if they are not expressed in the same way.

3 Proposed solution

To address the problems identified in the previous section, our approach focuses on four elements: first, we introduce the goal meta-model proposed to analyze the requests made by the user in structured natural language. Second, we propose our Ontologies needed to assist in resolving user queries. Third, we propose the use of annotation to specify the descriptor to be published in the registry. Finally, we propose the use of similarity metrics in our discovery algorithm. The elements of our proposed solution are discussed in the following paragraphs:

3.1 Goal meta-model

In this thesis, we focus on user needs expressed in structured natural language. We propose an ontology based solution for the interpretation of these needs. The implementation of the ontology requires the prior definition of a goal model for expressing queries. This model will allow the exploitation of ontology to find similarities between the elements of the user query and the attributes of the intentional services published in the registry.

The concept of basic goal we present in our work uses a lexical formalism with verb, target and parameters representing semantic functions of the verb. We support, in this context, the formulation of goals of *ISM* [9] based on a linguistic approach originally developed by [16, 17]. This approach inspired by the case grammar of Fillmore [6] and extensions [5] based on the fact that the semantics of goal is captured by a verb and parameters that correspond to roles associated to the verb. This formalism allows representing user goal and service intention (fig. 3). In this approach, a goal is expressed by a verb, a target and one or more parameters so-called 'direction', 'ways', 'time', 'beneficiary', 'quality', 'quantity' and 'location'. The verb and the target are mandatory while the parameters are optional. In general, any sentence can be expressed by Goal formalism.

This formalism allows representing user needs, and on the other hand, the intentions that intentional services can meet.

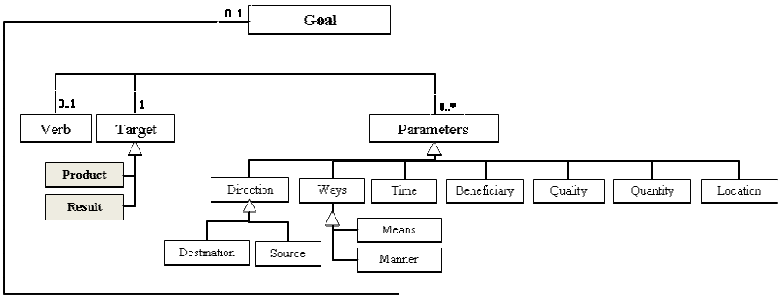

Fig. 3. *Goal meta-model for capturing user needs*

The formalism shows that the 'target' is either 'product' or 'result'. Result is non-tangible object and represents an output of the service. On the other hand, 'product' could be found as input and output.

3.2 Ontologies needed

We propose, from the intentional model, the ontologies needed to match the concepts of user queries with those of intentional services. Sharing the same ontologies allows establishing mappings at the time of service discovery.

However, queries are expressed by the users to find adapted services to solve their business needs. In practice, the needs are expressed as goals to be met. For example, finding a service to specify the requirements for booking a hotel room is reflected by the query "Book a hotel room". For that reason, we can write queries using ontologies and in particular each goal is expressed as a verb, a target and parameters. We differentiate two types of ontologies:

a) **Ontology of verbs** representing syntactic and semantic concepts related to verbs. It gives the different meanings of verbs and characterization of components used by these verbs in sentences of natural language. The specifications of the verb that agrees or refuses to construct sentences are syntactic concepts contained in the ontology. We rely on the classification of verbs in [20] to define this ontology.

b) **Ontology of products** defines a common vocabulary for all objects manipulated during the intentional services search [8]. This ontology is used especially to specify the inputs and outputs provided by the services.

The relation between these two ontologies defines the ontology needed for query resolution. In order to establish this relation, we related each concept in the ontology of products to a verb in the ontology of verbs.

3.3 Use of annotations

We propose to use an annotation approach to implement the intentional descriptions of services whose interpretation is made at the time of publishing the service in the registry (for cost reasons of and standards adoption).

The introduction of intentional perspective enriches the description of operational services. For this reason, we could use a semantic approach to annotate descriptors of Web services to enrich existing standards with intentional descriptions. We justify this choice by three simple reasons: (i) industries prefer updating their existing approaches to complete change, (ii) it is possible to annotate specific intentions using an ontology, which makes searching easier and (iii) from technical point of view: the annotations can be stored inside or outside the description. We propose the use of standard SAWSDL [23] to add intentional annotations then store them in the UDDI. This proposal is justified by [24] who also cited several reasons to show that SAWSDL directly supports the functional model (the formal description of the functionality of services) and data semantics (the formal description of the data exchanged between services). In this context, they demonstrated how SAWSDL can be used to achieve reuse, interoperability and agility. Furthermore, SAWSDL is an

independent approach to semantic representation languages thanks to the separation between the mechanism of semantic annotation and representation of semantic descriptions, which gives developers the flexibility to choose their semantic representation language, reuse models semantic domain, and annotate descriptions with multiple ontologies [2].

The main idea is to extend SAWSDL for enhancing expressiveness of service description. In SAWSDL, for a given WSDL element one can use many references to concepts in domain ontology but there is no specification of the semantic information nature: is it a verb, a target, a destination? etc. That is why we propose, in our description, a new attribute called *queryConcept* to give references to the query concepts corresponding in the same order, to the domain concepts listed in the original SAWSDL "*modelReference*" attribute. Indeed, our approach for intentional service description is based on the use of three types of ontologies. The first one contains only the concepts defining terms of query' concepts. The second type of ontologies is the verb Ontology; it is described and classified by Urrego [20]. The third type of ontologies is the product Ontology (Domain Ontology), which contains the semantics of the service domain products (e.g. travel).

In this way, we can define for each WSDL element two attributes. The first attribute, called *queryConcept*, references the corresponding concepts in query terms. The second attribute, called *modelReference*, contains a set of URI corresponding to the first list and which relay the Ontologies of verbs and products. Let's consider the example presented in Figure 4. The example presents the annotation of service named *book_hotel*. We can identify in the goal of this service the verb, the product and the destination. The importance of the extended attribute *queryConcept* is to distinguish the role of each term in the goal.

```
1   <?xml version = "1.0" ?>
2   <!DOCTYPE Intentional_Service SYSTEM "Service.dtd">
3   <Intentional_Service Code ="S_book_hotel">
4     <Atomic_Service>
5       <Interface
6                   serviceConcept="&QueryOntology;#verb &QueryOntology;#product
                    &QueryOntology; #destination“
7                   modelReference="&VerbsOntology;#book
                    &TourismOntology;#hotel  &TourismOntology;#France">
8                 <Intial_Situation> booking not made </Intial_Situation>
9                 <Final_Situation> booking made </Final_Situation>
10  </Interface>
11      …
```

Fig. 4. Example of intentional service descriptor

3.4 Use of similarity metrics

Many approaches have been proposed in various fields to measure the similarities: information systems engineering through the reuse of components, software

engineering to ensure the traceability of software code in the context of literature to index and retrieve documents, or even for the analysis of patterns of heterogeneous databases [25, 26, 27, 28, 29, 30, 31]. These different approaches provide metrics expressed as formulas more or less adapted to different situations. Most of them particularly interested in measuring the similarity between two texts, others look for similarities between the structured models.

Our position is that the mapping must be performed using metrics appropriate to the situation. Two specific criteria allow us to define such situations: first analysis of similarity between simple elements (such as verbs, products …), on the other hand, the analysis of similarity between complex elements (such as models of goals). Different types and properties of simple and complex elements can be identified, thus defining a complex type of metrics useful for measuring similarities [32].

A collection of such metric similarities can be exploited by a complex typology of similarity measures suitable for the analysis of similarities of the texts. Such a typology of similarity measures can be produced consistently and, as in our approach, used for mapping between users needs and services intentions.

5 Related work

Our work fits into the family of research approaches of goal-driven services. Most of these approaches [13, 22, 21, 4], focus on specifying goals in the context of searching Web services that meet these goals. In these approaches, different models have been proposed to specify goals without focusing on the problem of their capture. SATIS [12, 3], proposes ways to assist end users in the explanation of their intentions (goals). Moreover, the approach GODO [7] proposes models and tools to capture the goals of users with the help of an ontology and natural language. SATIS differs from this approach by proposing a process of incremental refinement of user needs to specify the characteristics of web services sought, as is the case in [10, 4]. And it differs from the latter approach in that it relies on models and techniques of semantic web to enrich the description of user needs and thus to suggest ways of reasoning and explanations of web services found to implement a business need.

Concerning the discovery of services, many logical approaches for the discovery of services [8] and hybrid matching algorithms have been proposed [11] using the metric of similarity. Other algorithms [2] make use of the service elements added to the interface. All these algorithms primarily exploit the degree of similarity by modifying the existing metrics. Classification mechanisms have been proposed too [1, 11, 2].

5 Conclusion

This work is in the research context to publish and discover services in a registry said intentional. We intend to design a comprehensive registry that meets the needs of business agents and the goal-driven discovery of these services.

The problem addressed in this article concerns querying and finding of intentional services. A user expresses his needs in natural language in a query with a goal. For this reason, we proposed a model aims to formulate these queries. This model will

allow the exploitation of ontologies to find matching and similarities between the concepts of goals made by user requests and intentions of those services.

The result of our work is a goal model and a proposal for new descriptor to be published in the registry. Finally, our future work will be the proposition of an algorithm to make the matching between the concepts of user query and those of intentional services published in the registry.

References

1. Benatallah, B., Hacid, M-S., Rey, C., Toumani, F.: Request rewriting-based Web service discovery. In Proceedings de International Semantic Web conference (ISWC 2003), pp 242-257, (2003)
2. Chabeb, Y., Tata, S., Ozanne, A.: YASA-M: Semantic Web Service Matchmaker. International Conference on Advanced Information Networking and Applications Perth, Australia, pp. 20-23 (2010)
3. Corby, O., Faron-Zucker, C., Mirbel, I. : Démarches sémantiques de recherche d'information sur le Web. In Proc. Ingénierie des Connaissances IC'09, Hammamet, Tunisia (2009)
4. Da Silva Santos, L. O. B., Guizzardi, G., Pires, L. F., van Sinderen, M.: From User Goals to Discovery and Composition. Advances in Conceptual Modeling - Challenging Perspectives. Lecture Notes in Computer Science 5833/2009. Springer Verlag, Berlin, pp. 265-274 (2009)
5. Dik, S.C.: The theory of functional grammar. Foris Publications, Dodrecht, Holland (1989)
6. Fillmore, C.J.: The case for case. In Universals in linguistic theory. Holt, Rinehart and Winston, Inc, E.Bach/R.T.Harms (eds) (1968)
7. Gomez, J., Rico, M., Garcia-Sanchez, F.: GODO: Goal Oriented Discovery for Semantic Web Services. 5th international Semantic Web Conference (2006)
8. Guzélian, G.: Conception de systèmes d'information: une approche orientée service. PhD thesis, Université de Paul Cézanne d'Aix-Marseille III, (2007)
9. Kaabi, R.S.: Une Approche Méthodologique pour la Modélisation Intentionnelle des Services et leur Opérationnalisation. PhD thesis, Université de Paris 1. (2007)
10. Kaabi, R. S., Souveyet, C.: Capturing Intentional services with Business Process Maps. Research Challenges in Information Science (RCIS), Ouarzazate, Morocco, (2007)
11. Klusch, M., Fries, B., Khalid, M., Sycara, K.: OWLS-MX: Hybrid OWL-S Service Matchmaking. Web Semantics: Science, Services and Agents on the World Wide Web Volume 7, Issue 2, (2009)
12. Mirbel, I., Crescenzo, P.: Des besoins des utilisateurs à la recherché de services web : une approche sémantique guide par les intentions. Rapport de recherche ISRN I3S/RR-2009-12-FR, (2009)
13. Norton, M. S. B.: A Refined Goal Model for Semantic Web Services. Second International Conference on Internet and Web Applications and Services (ICIW'07) (2007)
14. Papazoglou, M. P.: Service-Oriented Computing: Concepts, Characteristics and Directions. Proceedings de the 4th International Conference "Web Information Systems Engineering" (WISE'03). Rome, Italy, pp: 3-12 (2003)
15. Papazoglou, M. P., Georgakopoulos, D.: Service-Oriented Computing. Communication of the ACM Journal, [46]10: 24-28, (2003)
16. Prat, N.: Goal formalization and classification for requirements engineering. Proceedings of the Third International Workshop on Requirements Engineering: Foundations of Software Quality REFSQ'97, Barcelona, (1997)

17. Prat, N.: Réutilisation de la trace par apprentissage dans un environnement pour l'ingénierie des processus. PhD thesis, Université Paris1 (1999)
18. Rolland, C., Kaabi, R. S., Kraeim, N.: On ISOA: Intentional Services Oriented Architecture. International Conference on Advanced information Systems Engineering (CAISE), Springer-Verlag, Trondheim, Norway, (2007)
19. Rolland, C., Souveyet, C., Kraiem, N.: An Intentional View of Service-Oriented Computing. Revue Ingénierie des Systèmes d´Information (ISI), RSTI (Revue des Sciences et Technologies de l´Information). ISI 13[1]: 107–137 (2008)
20. Urrego, G. S.: Approches linguistiques pour l'ingénierie de besoins fonctionnels et non-fonctionnels. PhD thesis, Université de Paris 1 Panthéon-Sorbonne, 2005.
21. Vukovic, M., Robinson, P.: GoalMorph : Partial Goal Satisfaction for Flexible Service Composition. International Conference on Next Generation Web Services Practices (2005)
22. Zhang, K., Li, Q., Sui, Q.: A Goal-driven Approach of Service Composition for Pervasive Computing. 1st International Symposium on Pervasive Computing and Applications (2006)
23. Farrell, J., Lausen, H.: Semantic Annotations for WSDL and XML Schema. W3C Recommendation http://www.w3.org/TR/2007/REC-sawsdl-20070828/. (2007)
24. Verma, K., Sheth, A.: Semantically Annotating a Web Service. IEEE Internet Computing. vol. 11, no. 2, pp. 83-85. (2007)
25. Bianco, G., De Antonellis, V., Castano, S., Melchiori, M.: A Markov Random Field Approach for Querying and Reconciling Heterogeneous Databases. Proceedings of the 10th International Workshop on Database and Expert Systems Applications (DEXA'99), Florence, Italy. (1999)
26. Castano, S., De Antonellis, V., Zonta, B.: Classifying and Reusing Conceptual Components. Proceedings of the 11th International Conference on Conceptual Modelling (ER'92), pp. 121-138, Karlsruhe. (1992)
27. Castano, S., De Antonellis, V.: A Constructive Approach to Reuse of Conceptual Components. Proceedings of Advances in Software Reuse: Selected Papers from the Second International Workshop on Software Reusability. Lucca, Italy, pp.: 24-26. (1993)
28. Jilani, L.L. Mili, R. A.: Approximate Component Retrieval: An Academic Exercise or a Practical Concern? Proceedings of the 8th Workshop on Institutionalising Software Reuse (WISR8), Columbus, Ohio (1997)
29. Natt och Dag, J., Regnell, B., Carlshamre, P., Andersson M., Karlsson, J.: Evaluating Automated Support for Requirements Similarity Analysis in Market-driven Development. In Proceedings of REFSQ'01, 7th International Workshop on Requirements Engineering : Foundations of Software Quality. Interlaken, Switzerland, (2001)
30. Bouchachia, A. and Mittermeir, R.T.: Coping with Uncertainty in Software Retrieval Systems. Proc. of the 2nd Intl. Workshop on Soft Computing Applied to Software Engineering (SCASE.01). pp. 21-30, (2001)
31. Salton, G. and McGill, M. J.: Introduction to Modern Information Retrieval. McGraw-Hill.: 118-156, (1983)
32. Zoukar, I.: MIBE : Méthode d'Ingénierie des Besoins pour l'implantation d'un progiciel de gestion intégré (ERP). PhD thesis, Université de Paris 1 Panthéon-Sorbonne, (2005)

Integrating the Semantics of Events, Processes and Tasks across Requirements Engineering Layers

Maxime Bernaert (Supervisor: Geert Poels)

Department of Management Information Systems and Operations Management
Faculty of Economics and Business Administration,
Ghent University, Tweekerkenstraat 2, B-9000 Ghent, Belgium
Maxime.Bernaert@ugent.be

Abstract. Today, software should be more flexible, adaptable and more cost effective than ever before. There are indications that event-based architectures improve the flexibility, adaptability and cost effectiveness of software. Events are crucial concepts in event-based architectures, however, the concept of event has different interpretations in modeling techniques, which makes it difficult to integrate the use of different techniques during early and late requirements engineering. This paper outlines a PhD intended to develop an event-based requirements engineering methodology which supports the specification, development and verification of event-based systems. More specifically, this PhD strives to further develop the concept of event in requirements engineering and provide it with a formally defined semantics. The event concept is positioned with respect to existing concepts for modeling dynamic aspects of a system. A major goal is to keep the complexity of the modeling method at an acceptable level and enable a smooth transition of event-based architectures from requirements to implementation level. Finally, by performing an ontological analysis, using the BWW ontology and UFO, a set of orthogonal dimensions of the concept of event could be found.

Keywords: Requirements Engineering, Enterprise Ontology, Events, UFO, BWW

1 Introduction

Today, software should be more flexible, adaptable and more cost effective than ever before. Traditionally, interaction between components of a computer system is based on a mechanism of request/reply. Although the limitations of this interaction mechanism have been known for long, even the most recent technology platforms are still using request/reply as the basic principle for communication between a service provider and service consumer. Flexibility and adaptability of software can be significantly improved if an event-based architecture is applied [1] [2].

The power of event-based systems lies in the decoupling of sender and recipient which results in highly-flexible, adaptable and loosely-coupled systems. The majority

of the applications of event-based systems is realized by using events at the implementation level, for example by means of event-based middleware technology. Although the basic components of an event-based system (publishers, subscribers, events, subscriptions, consumers and producers) are known in middleware technologies under different names (notifications, dispatchers, publications and broadcasters) with different semantics, existing taxonomies [3] [4] give a quite clear picture of existing definitions and interpretations of events at implementation level.

In the software development process, the implementation phase is generally preceded by a phase of requirements engineering and design. In requirements engineering two phases can be distinguished: an early requirements phase, in which the required functionality of the system is explored with respect to the objectives, and a late requirements phase, in which the specifications of a system are modeled in detail. Although these phases are crucial in developing the core requirements of the system, they are lacking a full integration of event-based systems. As an example, the definition of events and actions in UML 2.0 remains strongly implementation oriented and UML does not provide clear definitions of events across abstraction levels and requirement engineering layers.

The absence of the event-based architectural style in requirements engineering significantly hampers the implementation of these systems. In addition it becomes difficult to examine the correctness of the developed systems. Often it will be almost impossible to make the automatic translation of event-agnostic platform independent models to event-based platform specific models [5].

The aim of this PhD is the development of an event-based requirements engineering methodology which supports the specification, development and verification of event-based systems. More specifically, this PhD strives to further develop the concept of event in requirements engineering and provide it with a formally defined semantics. The event concept is hereby clearly positioned with respect to and is related to existing concepts for modeling dynamic aspects of a system. A major goal is to keep the complexity of the modeling method at an acceptable level [6] and enable a smooth transition of event-based architectures from requirements to implementation level.

Today, a smooth transition between a conceptual data model, towards the logical data model and the physical data models has been established. This PhD aims at developing a similar framework for the dynamic aspects of a system. The emphasis lies on early and late requirements and the transition between the phases. Verifying the feasibility of the transition to an implementation architecture will serve as a validation of the research results.

1. A first objective is to make an ontological analysis (using upper level ontologies, in particular UFO and BWW) of the concept of event and related concepts like activities, tasks and (sub-) processes in the different requirements engineering phases. The aim is to distinguish different dimensions regarding the concept of event, in which existing modeling techniques can be positioned with respect to their use of events. A systematic review will be performed, according to the procedures outlined in [7].
2. In a second part of this PhD, a framework and meta-model are developed for modeling dynamic and interaction aspects in an event-based matter. In this meta-model the concept of event is defined and its relationship with other concepts is

clarified. Existing modeling standards (such as UML and BPMN) are evaluated for their use of the concept of event. Particular attention goes to the support of the meta-model for quality aspects of event-based requirements. The meta-model must offer support for both a preliminary model (in which not all consistencies and quality requirements must be satisfied) and for a final model (which must satisfy all quality requirements).

3. In the final part of this PhD, the meta-model is extended with guidelines and methods to ease the use of this meta-model.

This paper is structured as follows. Section 2 starts with explaining the methodology, followed by section 3, briefly describing the scientific contribution and innovative aspects. Section 4 introduces ontological research, section 5 and 6 give a short description of UFO and BWW respectively, after which section 7 provides us with an overview of the results achieved so far. Section 8 stipulates the future research directions and the paper ends with a conclusion in section 9.

2 Methodology

2.1 Ontological Analysis

The first objective is a further in-depth analysis of how the concept of event is used in the different requirement engineering layers and how its semantics have (not) been defined. Through previous research, a good knowledge of the use and semantics of the concept of event in industry standards such as UML and BPMN has been developed. However, an additional study of philosophically-based ontologies for information and knowledge systems and their representations (e.g. Bunge-Wand-Weber [8] and UFO [9]) can provide additional insights.

2.2 Developing the Meta-model

A meta-model provides a definition of the concepts and their relationships which enables traceability between the different layers and models. Traceability is an important feature of the meta-model as aggregation and causal relationships are cornerstone concepts when modeling behavioral aspects. Activities at a higher level are realized by means of a number of activities at a lower level. In this respect, one can speak of aggregation (bottom-up) or vertical causality (top-down): an action at a higher level causes activities at the lower level. The other way round: a pattern of events at the lower level can lead to the conclusion that an event has happened at a higher layer.

Furthermore, the problem of vertical causality and aggregation is strongly related to the issue of granularity of software services, an issue for which there is no satisfying answer yet.

2.3 Making Models Executable

Modeling languages that are executable or can easily be translated to an executable format (such as the translation of BPMN to BPEL) have significant advantages over non-executable modeling languages. The meta-model will therefore be enriched with executable semantics by means of process algebra.

3 Scientific Contribution and Innovative Aspects

Powerful modeling languages with precise semantics are a basic requirement to leverage the promises of Model Driven Engineering. This requires however that requirement modeling languages build on the best practices of implementation technologies and architectures. The innovative contribution of the proposed research is at the level of requirements engineering by enriching existing modeling languages with the power of event-based software development.

The outcome of this research is a meta-model that defines and relates the semantics of events, actions, processes and other related concepts in an unambiguous way and a method to use this meta-model. In addition, a mapping to existing modeling languages and standards such as UML and BPMN is defined.

4 Ontological Analysis

One of the most-cited definitions of ontologies is: "An *ontology* is an explicit specification of a conceptualization" (199, [10]). Ontologies should be regarded as descriptive models [5], representing reality by a set of concepts, their interrelations, and constraints under open-world assumption [11], which states that anything not explicitly expressed by an ontology is unknown. Upper level ontologies, like UFO and BWW, provide basic concepts for classification and description. By looking at ontologies to make an analysis of the concept of event, we go back to the roots in principled philosophical theories about what kinds of things exist and what are their basic relationships with each other. We adopt the position that concepts should be founded on an upper level ontology referring to reality in a philosophically justified way. We assess goodness in terms of how well information systems embody the meaning of the real-world system they are intended to model [8], that is why it is crucial to look at the meaning of the concept of event in ontologies.

5 Unified Foundational Ontology (UFO)

In [9], a foundational ontology named UFO (Unified Foundational Ontology) has been developed, which can be used as a theoretically sound basis for evaluating and redesigning conceptual modeling languages in general, and ontology representation languages in particular. UFO is derived from a synthesis of two other foundational

ontologies, GF0/GOL and OntoClean/DOLCE. The main purpose of UFO is to provide a foundation for conceptual modeling, including business modeling.

UFO addresses issues such as: the general notions of types and their instances; objects, their intrinsic properties and property-value spaces; the relation between identity and classification; distinctions among sorts of types and their admissible relations; distinctions among sorts of relational properties; part-whole relations.

UFO is divided into three incrementally layered compliance sets: UFO-A defines the core of UFO, excluding terms related to perdurants and terms related to the spheres of intentional and social things; UFO-B defines, as an increment to UFO-A, terms related to perdurants; UFO-C defines, as an increment to UFO-B, terms related to the spheres of intentional and social things, including linguistic things. In particular UFO-B discusses the meaning of the concept of event. UFO-C talks about intentional and social concepts.

6 Bunge-Wand-Weber (BWW)

The BWW ontology, a framework created by Wand and Weber [8] on the basis of the original metaphysical theory developed by Mario Bunge [12] [13], is developed to model information systems. They argue that they have relied on Bunge's ontology for three reasons [14]. First, they contend that Bunge's ontology is better developed and better formalized than any competing ontology they have encountered. Second, Bunge models the world as a world of systems. Bunge uses concepts that are fundamental to the computer science and information systems domains. Third, Wand and Weber argue they have been able to produce useful results using Bunge's model.

Having chosen Bunge's model as the basis for their work, Wand & Weber [8] argue the BWW model can be used to understand and predict the characteristics of good information systems grammars. Good is defined in a restricted way to indicate how well the scripts produced using the grammar convey the deep structure or meaning of the real-world system the information system is intended to represent.

By starting to look at ontologies to derive the meaning of the concept of event, we believe a good analysis can be made of existing meanings of the concept of event in requirements engineering. The position defended here is that, in order to model reality, the concepts used (like events) should be founded on upper level ontologies.

The purpose of the ontological model Wand and Weber have proposed is to define a set of constructs that are necessary and sufficient to describe the structure and behavior of the real world [8]. This set of constructs provides a benchmark to evaluate whether those grammars used to describe real-world systems are ontologically complete.

7 The Proposed Approach and the Results Achieved so far

A development process consists of different phases (conceptual modeling, design and implementation), which should fulfill different sets of requirements. This PhD is

concerned with the meaning of the concept of event in conceptual modeling, in particular in the early and late phases in software engineering.

Conceptual modeling's main objective is concerned with identifying, analyzing and describing the essential concepts (for instance the concept of event) and constraints of a universe of discourse with the help of a modeling language that is based on a set of basic modeling concepts (forming a meta-model). Conceptual models therefore have to resemble the concepts of the real world in the most appropriate way, therefore our basis to evaluate different modeling languages will be the use of the concept of event in upper level ontologies, in particular UFO and BWW.

7.1 UFO

According to UFO, the concept of event can be classified according to three orthogonal dimensions: atomic event – complex event, instantaneous event – time-extended event, action event – non-action event.

7.1.1 Atomic Event – Complex Event

UFO-B makes a distinction between atomic events and complex events [15] [16].

An atomic event is an event that has no improper parts. Examples: an explosion, a message reception. A complex event is an event that is an aggregation of at least two events (that can themselves be atomic or complex). Examples: a parallel occurrence of two explosions, an absence of a message reception (without some time window).

A process is a complex event that is a sequence of two or more (possible parallel occurrences of) atomic events. Examples: a storm, a football game.

7.1.2 Instantaneous Event – Time-Extended Event

Does the event take place without duration or is there a time duration involved?

In [16], the authors state that it is important to emphasize that being atomic and being instantaneous are orthogonal notions in this framework, i.e., atomic events can be time-extended as well as an instantaneous event can be composed of multiple (instantaneous) events.

7.1.3 Action Event – Non-action Event

In UFO-C, a distinction is made between action events and non-action events [15].

An action event is an event that is created through the action of a physical agent. A non-action event is an event that is not created through an action of a physical agent.

A physical agent is a physical object that creates action events affecting other physical objects, that perceives events, possibly created by other physical agents, and to which we can ascribe a mental state. Examples: a dog, a human, a robot.

Action events are intentional events [16], i.e., events which instantiate a plan with the specific purpose of satisfying some intention.

7.2 BWW

BWW describes three dimensions of the concept of event: event – process, internal event – external event, well-defined event – poorly-defined event.

7.2.1 Event – Process

A BWW-event is "A change of state of a thing. It is effected via a transformation" [8].

A BWW-process is "An intrinsically ordered sequence of events on, or states of, a thing" [17]. "Processes are either chains or trees of events" [12].

7.2.2 Internal Event – External Event

A BWW-internal event is "An event that arises in a thing, subsystem or system by virtue of lawful transformations in the thing, subsystem or system. The before-state of an internal event is always unstable. The after-state may be stable or unstable" [8].

A BWW-external event is "An event that arises in a thing, subsystem, or system by virtue of the action of some thing in the environment of the thing, subsystem or system. The before-state of an external event is always stable. The after-state may be stable or unstable" [8].

A BWW-stable state is "A state in which a thing, subsystem or system will remain unless forced to change by virtue of the action of a thing in the environment (an external event)" [8].

A BWW-unstable state is "A state that will be changed into another state by virtue of the action of transformation in the system" [8].

7.2.3 Well-defined Event – Poorly-defined Event

A well-defined event is an event in which the subsequent state can always be predicted given that the prior state is known [8]. A poorly-defined event is an event in which the subsequent state cannot be predicted given that the prior state is known [8].

8 Future Research

Starting from the different dimensions of the concept of event found in UFO and BWW as upper level ontologies, a first step is set to investigate modeling techniques, for example BPMN, BPDM, UML, and their use of the concept of event. By performing a systematic review with a clearly defined protocol, a list of modeling techniques will be distilled in which the concept of event is used. Our intention is to make a mapping of the different existing modeling techniques, according to their use of the concept of event, using the different dimensions found in UFO and BWW. This ontological mapping will give us a clear view of the most frequently used meaning of the concept of event and the least frequently used meaning of the concept of event.

This ontological analysis will be a good starting point to start developing an ontologically well-based meta-model, where the concept of event is used in all its possible meanings. The meta-model can be developed starting from existing meta-

models, or can be developed from scratch, if no sufficient basis of meta-models is available.

This meta-model will be supported by a method, which provides us with guidelines on how to interpret the meta-model and develop models.

Complementary to an ontological analysis, a study starting from existing modeling techniques can distill the meaning of the concept of event as used in these modeling techniques. Comparing the ontological analysis with this research can provide us with interesting new insights.

9 Conclusion

The concept of event has several interpretations. By making an ontological analysis, based on UFO and BWW, two sets of each three orthogonal dimensions have been discovered on which existing modeling techniques can be mapped according to their use of the concept of event.

UFO makes a distinction between atomic and complex events, instantaneous and time-extended events and action events and non-action events.

BWW's dimension of event and process resembles the atomic-complex dimension of UFO. The two other dimensions, internal event – external event and well-defined event – poorly-defined event, are two other dimensions.

By projecting the used event semantics onto these ontological dimensions, it becomes easier to compare different modeling techniques. If we want to make a meta-model, we need to have a clear understanding of the concepts used in the existing models.

References

1. Mühl, G., Fiege, L., Pietzuch, P.: Distributed Event-Based Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
2. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Rule Representation, Interchange and Reasoning on the Web (2008) 3-3
3. Barrett, D.J., Clarke, L.A., Tarr, P.L., Wise, A.E.: A framework for event-based software integration ACM Transactions on Software Engineering and Methodology Vol. 5 (Oct. 1996) pp. 378-421
4. Meier, R., Cahill, V.: Taxonomy of Distributed Event-Based Programming Systems. The Computer Journal 48 (2005) 602-626
5. Aßmann, U., Zschaler, S., Wagner, G.: Ontologies, Meta-models, and the Model-Driven Paradigm. Ontologies for Software Engineering and Software Technology (2006) 249-273
6. Siau, K., Erickson, J., Lee, L.Y.: Theoretical vs. Practical Complexity: The Case of UML Journal of Database Management Vol. 16 (July-Sept. 2005) pp. 40-57
7. Kitchenham, B.: Procedures for performing systematic reviews. Technical Report TR/SE-0401 (2004)
8. Wand, Y., Weber, R.: Towards a theory of the deep structure of information systems. In DeGross, J., Alavi, M. & Oppelland, H. (eds), Proceedings of the Eleventh International Conference on Information Systems, Copenhagen (1990b) pp. 61-71

9. Guizzardi, G.: Ontological foundations for structural conceptual models. Enschede (2005) 416
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowl. Acquis. 5 (1993) 199-220
11. Horrocks, I., Schneider, P., van Harmelen, F.: From SHIQ and {RDF} to {OWL}: The making of a web ontology language. Journal of Web Semantics 1 (2003) 7-26
12. Bunge, M.: Treatise on Basic Philosophy, Vol. 3, Ontology I: The Furniture of the World. Reidel, Boston (1977)
13. Bunge, M.: Treatise on Basic Philosophy: Volume 4: Ontology II: A World of Systems. Reidel, Boston (1979)
14. Wand, Y., Weber, R.: On the ontological expressiveness of information systems analysis and design grammars. Information Systems Journal 3 (1993) 217-237
15. Guizzardi, G., Wagner, G.: Some applications of a Unified Foundational Ontology in Business Modeling. In: Green, P., Rosemann, M. (eds.): Business Systems Analysis with Ontologies. Idea Group Publishing (2005)
16. Guizzardi, G., Falbo, R.A., Guizzardi, S.S.: Grounding Software Domain Ontologies in the Unified Foundational Ontology (UFO): The case of the ODE Software Process Ontology. Iberoamerican Workshop on Requirements Engineering and Software Environments (IDEAS2008), Recife (2008)
17. Green, P.F.: An Ontological Analysis of Information Systems Analysis and Design (ISAD) Grammars in Upper CASE tools. Department of Commerce. University of Queensland (1996)

Towards a Framework for Business Process Models Reuse

Wassim Derguech

DERI, Digital Enterprise Research Institute,
National University of Ireland, Galway
firstname.lastname@deri.org
http://www.deri.ie

Abstract. Despite there exist different proposals for reusing existing well-designed artifacts for process modelling and capturing variability in business process models, most of them suffer from three major shortcomings: lack of automation support in (i) maintaining the configurable process model, (ii) modelling sub-processes as stand alone entities and (iii) user-centricity and decision support in choosing the suitable alternatives. In this context, we propose a framework that allows for reusing process models. We present in this paper our ongoing research in defining this framework that contains a data structure and its construction principles as well as first thoughts about maintaining it when adding process variants.

Key words: Business process, business process modelling, configurable process modelling.

1 Introduction

Process Aware Information Systems (PAISs) [1] are used to manage and execute operational processes involving people, applications and data sources on the basis of business process models. The discipline that is concerned by this process-centric trend is known as Business Process Management (BPM) [2]. BPM includes concepts, methods, and techniques to support the design, administration, configuration, enactment, and analysis of business processes [2]. The basis of BPM is the explicitness of business processes presented by process models.

A business process model is a graphically designed network based on symbols that represent procedures or activities [3] which collectively describe how an organization conduct its business processes .

Business process modelling phase, requires intensive knowledge related to both (i) the business domain and (ii) the modelling language. In fact, the business modeler has to define exactly what tasks must be captured, their execution logic as well as how to specify them through a modelling language. Designing high quality process models from scratch is often time consuming, error-prone and costly [4]. Thus, sharing and reusing business process models by different business actors (e.g., modelers, managers, developers) is highly recommended.

1.1 Motivation

The advent of Reuse-Oriented Development (ROD) in BPM brings a number of reference models and repositories of process templates. They are used to design business process models exploiting proven practices [5, 6]. The benefits of reuse in PAISs are widely accepted in theory and practice [7]. The idea is to generate a process model from already well-designed artifacts. Yet, proposed solutions have not been proved as effective as expected. They are either not supporting enough automation to meet frequent changes in process variants, or they require advanced modelling experience. In other words, they are not applicable without strong modelling language background or support. In the whole paper, we use user experience to refer to modelling language skills.

In addition, configurable process models enable to capture variants of large process models. However, they do not allow for modelling sub-processes within these large models. A customizable process model that manages both processes and sub-processes is needed.

At a first step of this research, we have identified the following requirements for reusing business process models:

1. *Resource efficiency*: First, there is a need to have an approach that enables managing process variants in an efficient way. By efficient way we mean overtaking both resource redundancy and inconsistency problems.
2. *Maintainability*: Second, an approach should support automatic changes. By support automatic changes we refer to adding or removing a variant from the configurable process model with an automation support.
3. *Modularity*: Third, an approach should allow for modelling both processes and their sub-processes.
4. *User-centricity*: Fourth, an approach should not require much modelling skills from the end users.

1.2 Research Problem

The research goal we are trying to reach in this thesis is summarized as follows:

> *Develop an efficient approach that allows for reusing existing process models without (or with little) modelling experience. Our approach should offer a configurable model, provide an automation support for its maintainability and allow for modelling both processes and sub-processes.*

To manage configurable process modelling in an efficient way, we suggest providing a framework that captures the different process variants. A process variant, as it is understood here, is a single process model that achieves a given goal. Such a framework should be easily updated via an automation support. It should also enable business process modelers to define the most suitable (sub)process variant that suits their requirements.

1.3 Outline

The rest of this paper is divided into three sections. Section 2 shows how current solutions handle business process model reuse. Section 3 provides our preliminary work in defining a framework to represent a configurable process model. And finally section 4 concludes the paper and gives brief overview of our future work.

2 Related Work

Several solutions have been investigated to support the reuse of business process models. We classify these solutions into two categories: (i) discovery of existing models and (ii) adaptation of a structure that captures possible model's variants.

2.1 Discovery of existing models

Under this first category, we distinguish two possible approaches. The first one consists of discovering an entire business process model by querying a repository and selecting the most suitable one [8–10]. This method suffers from resource redundancy because it does not consider common parts of process model's variants and consequently they are duplicated in each entry of the repository. In addition, if any changes must be applied on the common parts of process variants, all the entries must be updated and from this arises the inconsistency problem.

The second approach overcomes the problem of resource redundancy and inconsistency by considering business process building blocks instead of entire models. This approach consists of discovering these building blocks and aggregating them in order to construct a business process model [11]. However, modelling experience is needed. In fact, the modeler has to be familiar with these business process building blocks and manage their aggregation for creating a complete model.

2.2 Adaptation of a structure that captures possible model's variants

Here we consider solutions that aim at merging different variants of a business process model into a single model. This is suitable for avoiding both resource redundancy and inconsistency problems. Under this category, we distinguish two possible approaches.

The first one provides a process model which contains placeholders that need to be detailed during the modelling phase [11]. These placeholders represent places where differences between process variants occur. During the modelling phase, users need to rely on their modelling experience to refine the placeholders.

The second approach manages configurable business process models [12]. A configurable model is the result of merging model's variants into a single model. In this case, the modelling phase consists of enabling or disabling different branches of the configurable process model.

This approach has been extended by "Questionnaire-driven Configuration of Reference Process Models" [13]. Here authors introduced a user-friendly method for customizing configurable process models. In fact, the user specifies his business requirements by answering a set of domain-related questions that are mapped to the configurable process model.

However, a process modeler has to define the configurable process model and sit with the domain expert in order to define domain constraints and their mapping to the configurable model. Thus, adding a new variant in this framework, requires another meeting between the modeler and the domain expert in order to add a variant to the process model, define its mapping with the domain constraints and update the questionnaire model. Even if this solution is suitable for a user-centric modelling, it needs to be improved to support automation maintainability of both configurable process model and questionnaire model.

In addition, configurable process models presented in [12, 13] enable to capture variants of large process models. However, they do not allow for modelling sub-processes within these large models.

3 A Framework for Managing Process Variants

To cope with the aforementioned research problem, we introduce a framework to represent configurable process models. This framework consists of a data structure, a set of structuring principles and maintaining operations. At this stage we show how to maintain this data structure only when adding a new variant. In this first version of our study, we consider a business process model as a sequence of business goals.

3.1 Data structure Definition

Our framework defines a data structure as a tuple CPM=$\{\Sigma,\Gamma,\Delta\}$ where:

- Σ represents the set of business goals involved in the whole configurable business process model,
- Γ represents the set of abstract business goals (will be introduced later in this paper) and
- Δ represents the possible variants of each business goal that can be a sequence of business goals. Entries of Δ are presented as:
  $BusinessGoal_1 : BusinessGoal_2(-BusinessGoal_N)^*$ such that
  $\{BusinessGoal_1, BusinessGoal_2, ...BusinessGoal_N\} \subseteq \Sigma \cup \Gamma$
  and "$BusinessGoal_2 - BusinessGoal_3$" is the sequence between the two business goals (i.e., $BusinessGoal_2$ and $BusinessGoal_3$).
  This means that possible variants of BusinessGoal1 are presented as a sequence of other business goals. We call $BusinessGoal_1$ a *variation point* and $BusinessGoal_2(-BusinessGoal_N)^*$ a *variant*.

The framework should respect a set of constraints/principles to assure that the data structure will remain valid and well-formed after an update operation.

1. *Minimality*: Each element of Δ has to be defined only once and should not be derived from other elements of Δ.
2. *Coverage*: By necessity and nature, the framework must cover all defined variants.
3. Consistency: Only defined variants should be deduced from the framework.

3.2 Maintaining the data structure when adding a new variant

When building the data structure, we start from a set of variants and we add them, one at a time, assuring that the principles defined previously are not violated. When adding a new variant of a goal, the idea is to check, using matching detection, whether the variant (or parts of it) already exists in the data structure. There are three situations that may occur according to the matching degree between business goals of the new variant and those in the configurable process model.

1. *Perfect match:* In this case the current variant to be inserted is entirely found in the current data structure. In such situation there is no action to be taken and the data structure remains as it.
2. *No matching:* In this case the current variant to be inserted is not found in the current data structure, not even partially. The variant is inserted as follows: all business goals composing the variant are added to Σ and the variant description is added to Δ.
   Example: We want to insert the variant A:G-H-I in $CPM_1 = \{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
   - $\Sigma_1 = \{A, B, C, D, E, F\}$
   - $\Gamma_1 = \{\ \}$
   - $\Delta_1 = \{A:B\text{-}C, A:F\text{-}E, C:D\text{-}E\}$

   The updated configurable process model is then $CPM_1=\{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
   - $\Sigma_1 = \{A, B, C, D, E, F, G,H,I\}$
   - $\Gamma_1 = \{\ \}$
   - $\Delta_1 = \{A:B\text{-}C, A:F\text{-}E, C:D\text{-}E, A:G\text{-}H\text{-}I\}$

   Using a tree representation, the variant is inserted as an alternative of the variation point as shown in Fig. 1[1].

3. *Partial match:* An intermediary situation is when a partial match occurs between the process variant to be inserted and the current data structure. In this case we distinguish two possible situations:

   - The first situation occurs when the new variant has common parts with another variant of the same business goal. A typical example is depicted in Fig. 2. This example shows adding the variant A:B-H-I in $CPM_1 = \{\Sigma_1,\Gamma_1,\Delta_1\}$ where:
     - $\Sigma_1 = \{A, B, C, D, E, F, G\}$

[1] All figures in the paper are following the BPM notation: www.bpmn.org

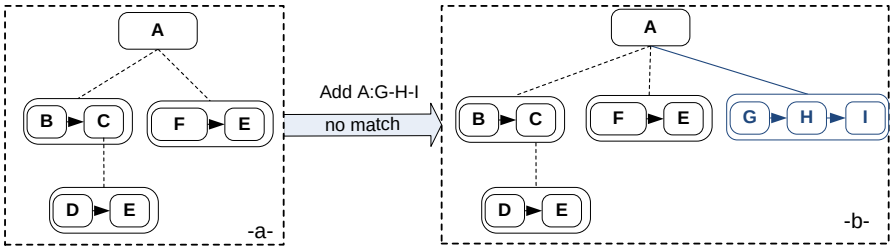

Fig. 1. The data structure (-a-) before and (-b-) after the insertion of a variant

- Γ_1 ={ }
- Δ_1 = {A:B-C, A:F-G, C:D-E}

The new variant A:B-H-I has B in common with A:B-C. To add this variant, an abstract business goal α is introduced to replace the different parts of these variants. The updated configurable process model is then CPM_1={Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E, F, G, H, I}
- Γ_1 ={ α}
- Δ_1 = {A:B-α, α:C, α:H-I,A:F-G, C:D-E}

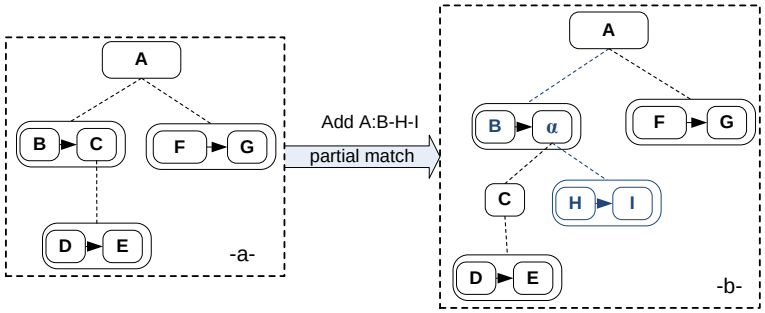

Fig. 2. The insertion of a variant including an abstract business goal

– The second situation occurs when the new variant has common parts with another variant but not of the same business goal. A typical example is depicted in Fig. 3. This example shows adding the variant A:D-F in CPM_1 = {Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E}
- Γ_1 ={ }
- Δ_1 = {A:B-C, B:D-E}

This situation is similar to the second case (no match) and the updated configurable process model is then CPM_1={Σ_1,Γ_1,Δ_1} where:

- Σ_1 = {A, B, C, D, E, F}

- $\Gamma_1 = \{\}$
- $\Delta_1 = \{A{:}B\text{-}C,\ B{:}D\text{-}E, A{:}D\text{-}F\}$

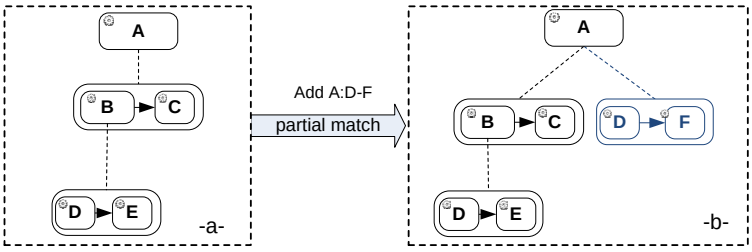

Fig. 3. The insertion of a variant with partial match without introducing an abstract business goal

4 Conclusion and future work

Reusing process models is an important concept for Business Process Management because it can decrease the modelling time and reduce the business user's work and risk to make errors.

Several solutions have been provided for process model reuse. Despite their benefits, they have not yet been adopted to capture configurable process models. This is mainly due to the issues such as the lack of automation support in handling changes (i.e., adding or removing process variants).

In this thesis we propose a framework for managing configurable process models. It defines a data structure that captures process model variability at the business goal level. We presented a set of principles that the proposed data structure has to comply with and we show how it is maintained when adding a new variant.

Our work is still in an early stage and continuous improvements are planned as a future work:

- In the near future, we plan to formally define construction principles. Further construction principles could be considered as well.
- A number of maintaining operations have not yet been explored or are still under definition, for example the deletion of a variant.
- We intend to investigate and extend this framework in order to consider other block patterns (we have presented only sequence pattern in this paper).
- From exploitation point of view, we intend to capture configuration parameters at each variation point. These parameters will help for generating a question flow to assist users in defining their process models.
- Our framework is not exclusively designed for managing process variability. We plan to experiment our approach in the context of Mashups applications development.

Acknowledgments

This thesis is supervised by Dr. Sami Bhiri from the Service Oriented Architecture Unit, National University of Ireland Galway, Digital Enterprise Research Institute. This work is funded by the Lion II project supported by Science Foundation Ireland under grant number 08/CE/I1380.

References

1. Dumas, M., van der Aalst, W.M., ter Hofstede, A.H., eds.: Process-Aware Information Systems : Bridging People and Software through Process Technology. Wiley-Interscience, Hoboken, NJ (2005)
2. Weske, M.: Business Process Management: Concepts, Languages, Architectures. Springer (2007)
3. van der Aalst, W.M.P.: Business process management. In Liu, L., Özsu, M.T., eds.: Encyclopedia of Database Systems. Springer US (2009) 289–293
4. Indulska, M., Green, P., Recker, J., Rosemann, M.: Business process modeling: Perceived benefits. In Laender, A.H.F., Castano, S., Dayal, U., Casati, F., de Oliveira, J.P.M., eds.: ER. Volume 5829 of Lecture Notes in Computer Science., Springer (2009) 458–471
5. Fettke, P., Loos, P.: Multiperspective evaluation of reference models - towards a framework. In Jeusfeld, M.A., Pastor, O., eds.: ER (Workshops). Volume 2814 of Lecture Notes in Computer Science., Springer (2003) 80–91
6. Fettke, P., Loos, P., Zwicker, J.: Business process reference models: Survey and classification. In Bussler, C., Haller, A., eds.: Business Process Management Workshops. Volume 3812. (2005) 469–483
7. Fettke, P., Loos, P.: Classification of reference models: a methodology and its application. Information Systems and E-Business Management **1**(1) (January 2003) 35–53
8. Lu, R., Sadiq, S.W.: On the discovery of preferred work practice through business process variants. In: Proceedings of Conceptual Modeling - ER 2007, 26th International Conference on Conceptual Modeling. (2007) 165–180
9. Markovic, I., Pereira, A.C.: Towards a formal framework for reuse in business process modeling. In Ter, Benatallah, B., Paik, H.Y., eds.: 2nd ed. of proceedings of the Workshop Advances in Semantics for Web services 2007 (semantics4ws 2007); contained in BPM 2007 Workshops, Brisbane, Australia, Springer Verlag (2008) 484–495
10. Markovic, I., Pereira, A.C., Stojanovic, N.: A framework for querying in business process modelling. In Bichler, M., Hess, T., Krcmar, H., Lechner, U., Matthes, F., Picot, A., Speitkamp, B., Wolf, P., eds.: Multikonferenz Wirtschaftsinformatik, GITO-Verlag, Berlin (2008)
11. Becker, J., Delfmann, P., Knackstedt, R.: Adaptive Reference Modeling: Integrating Configurative and Generic Adaptation Techniques for Information Models. In: Reference Modeling. Physica-Verlag HD (2007) 27–58
12. Rosemann, M., van der Aalst, W.M.P.: A configurable reference modelling language. Inf. Syst. **32**(1) (2007) 1–23
13. La Rosa, M., Lux, J., Seidel, S., Dumas, M., ter Hofstede, A.H.M.: Questionnaire-driven configuration of reference process models. In: Proceedings of the 19th International Conference on Advanced Information Systems Engineering, CAiSE. (2007) 424–438

Decision-Support for Service Bundling

Thomas Kohlborn

Faculty of Science and Technology, Queensland University of Technology,
126 Margaret Street, 4000 Brisbane, Australia
t.kohlborn@qut.edu.au

Abstract. Offering service bundles to the market is a promising option for service providers to strengthen their competitive advantages, cope with dynamic market conditions and deal with heterogeneous consumer demand. Although the expected positive effects of bundling strategies and pricing considerations for bundles are covered well by the available literature, limited guidance can be found regarding the identification of potential bundle candidates and the actual process of bundling. The proposed research aims at filling this gap by offering a service bundling method complemented by a proof-of-concept prototype, which extends the existing knowledge base in the multidisciplinary research area of Information Systems and Service Science as well as providing an organisation with a structured approach for bundling services.

Keywords: Service, service-orientation, bundling

1 Introduction

The interest in service-orientation has increased over the last years due to new technological developments [1] and novel approaches for organizational management [2] since services have become focal units for the cost-effective creation of customer value and innovation. The multidisciplinary nature of service-oriented concepts has led to the emergence of Service Science as a new academic discipline [3]. Business Service Management (BSM) can be positioned as the business discipline within Service Science dedicated to the holistic management of services in an organization to ensure alignment between the needs of the customer and the objectives of the organization [4].

Business Service Management is a research project as part of the Smart Services Cooperative Research Centre (CRC) research initiative.[1] An essential component of the strategic side of BSM is Service Portfolio Management (SPM) [5], which is one of the work packages within the BSM project and is led by Prof. Michael Rosemann, who is also the principle supervisor of the candidate's research.[2]

[1] Please refer to http://smartservicescrc.com.au

[2] Associate Supervisors are Dr. Axel Korthaus and Dr. Erwin Fielt. Both are postdoctoral research fellows at the Queensland University of Technology.

In focus of SPM is the service portfolio, which comprises a well-defined set of services. Offering innovative service bundles to the market is a promising option for service providers to strengthen their competitive advantages, cope with dynamic market conditions and heterogeneous consumer demand [e.g. 6]. Service bundles are composed of at least two services that can be integrated to a certain extend to create a new packaged service offering. Therefore, a key task within SPM is service bundling that deals with the challenge of identifying services within the portfolio that can and should be used to create service [5]. A demanding managerial task exists in comprehending potential service candidates offered internally and externally, and being able to identify service bundles that lead to new efficient and strategically-aligned packages.

Literature lacks approaches that facilitate the creation of adequate service bundles. Despite the fact that companies across all industry sectors with increased market pressures are challenged by the issue of service bundling [7], only little guidance has been provided so far for the identification of potential bundle candidates and for the actual process of bundling to answer the questions: "How can services be identified that should be bundled?" and "how can the act of service bundling be effectively and efficiently supported?".

The research related to the candidate's studies tries to answer these questions by developing a service bundling method. In particular, it will focus on the perspective of a service provider. Existing methods for service bundling usually use a given customer demand to drive the creation of service bundles [7-9]. While these methods are useful for situations where customer demand is well known and understood, poor performance can be expected when demand is hard to capture or anticipate. As customer-driven service bundles typically relate to an outside-in perspective, the induction of new, innovative bundles to the market to trigger demand is not yet sufficiently covered. The proposed method will fill this gap and provide an inside-out perspective on service bundling. The main contribution of the research is therefore a structured guideline to facilitate the composition of bundles in practice. Designed as an innovative artefact it extends the knowledge base of service management, while facilitating a multi-disciplinary approach honouring the importance of business and IT alignment. As a proof of concept, the method will be complemented by a software prototype.

The remainder of this paper is structured as follows. Based on the problem description that has been provided in this section, we first present related work and alternative approaches in this area of research, before the foundations of our approach will be detailed further. Subsequently, a detailed research design that aims at providing a structured guideline to answer the stated research questions will be described. Finally, the current status of the research is presented before the paper ends with a conclusion and directions for further research.

2 Related Work and Alternative Approaches

The objective of this research is to provide a service bundling method. A review of the academic knowledge base yields various possible approaches that can be utilized to identify service bundles as pointed out in [10].

For example, the area of artificial intelligence (AI) research offers techniques that can potentially support the design of solutions to the service bundle identification problem. Particularly, machine learning solutions are conceivable that can "learn" from existing successful service bundles to identify or propose new service bundles [e.g. 11]. A general problem of machine learning is that it usually does not yield absolute guarantees of the performance of algorithms. Moreover, in spite of many successes, AI research in general has been the target of fundamental criticism [e.g. 12]. To the best of the authors' knowledge, comprehensive AI approaches to identify and analyze new service bundles are not existent in the academic knowledge base.

Business Intelligence (BI) employs systems that "*combine data gathering, data storage, and knowledge management with analytical tools to present complex and competitive information to planners and decision makers"* [13]. Hence BI is used to analyze existing data to support future decisions. Within BI, the area of Association Rule Mining can be employed to identify bundle candidates. This mining approach analyses basket data type transactions, for example receipts from a supermarket, to identify items that are frequently bought together within one transaction [14]. The identified so called *frequent item sets* are used for recommender systems to offer customers related products, hence enabling cross-selling potentials (e.g. Amazon.com).

The ideas of Semantic Web approaches can be utilized as well. These approaches generally require three sorts of machine-understandable information: "*ontologies to define vocabulary, data about observations of the world, and theories that make predictions on such data"* [15]. An ontology specifies an explicit, simplified view of the world [16]. Using these artifacts, it is possible to model the relations between services and reason about them. This has been done for web services in the Web Service Modeling Ontology [17] and for real-world services using the OBELIX (Ontology-Based Electronic Integration of Complex Products and Value Chains) service ontology [7].

A different semantic approach utilizes Latent Semantic Indexing (LSI), also called Latent Semantic Analysis. Latent semantic indexing is an "*information retrieval technique based on the spectral analysis of the term-document matrix"* [18]. This is done through the creation of vector spaces using a mathematical technique called singular value decomposition. The created vector spaces can be queried for the semantic distance (usually expressed as a vector) between two terms or service descriptions, and the so found semantic relationships can be utilized to identify clusters or bundles of services. LSI requires no formal ontology that classifies the different elements of a service [19] and uses unstructured text documents as the only source of information. Several limitations to LSI restrict its usage, the main drawback being that the underlying vector space model "*is unable to convey any relationship [...] existing between the terms"* [20].

All of the introduced identification approaches can be expected to provide interesting results for a bundling method. As motivated in the next section, the

proposed method is based on service descriptions. Thus, a semantic approach has been chosen. Instead of employing a formal ontology, the notion of relationships is used to discover semantics between services. This ensures the simplicity of the method and its utility across a range of different service descriptions.

The single work that specifically targets service bundling is from Baida (please refer to [8]). The author used an ontology-based approach "*to facilitate the automation of the service bundling task"*. The created ontology includes the notion of resources as prerequisites or outcomes of service elements and so-called functions and relationships that define dependencies between two service elements (e.g. enhancing, excluding, and substituting). Using a given customer demand by expressing required resources, the method can create service bundles that satisfy the demand and adhere to the predefined set of dependencies between services. The author named his approach "Serviguration" to express his view of service bundling as a configuration task. A detailed discussion of the differences and distinguishing characteristics of the "Serviguration" approach and the proposed approach will follow in the next section.

3 Conceptual Framework for a Service Bundling Method

Preliminary Considerations

The purpose of the proposed method is the identification of possible service bundles. It is designed to support the process of bundle creation in its early stages. The method therefore focuses on limiting the solution space of possible bundles, using indicators that express some form of bundling motivation as described in [10].

It is important to point out that this method is not supposed to omit the evaluation of bundles by a domain expert. It has to be acknowledged that the domain expert is still needed to evaluate the overall feasibility of bundles, since this requires complex analysis, often utilizing tacit knowledge across a range of different disciplines (e.g. economy, marketing, legal). Rather, the aim of this method is to limit the scope of the necessary evaluation for the domain expert.

This approach is particularly useful when a large number of services are available, which is a common scenario particularly in business networks or service ecosystems [21]. Since a human expert would be overwhelmed by the task, the proposed method could be applied in an automated way using a corresponding support tool in order to quickly constrain the solution space of possible bundles. Consequently, the domain expert can focus on evaluating only the short-listed bundles that somehow indicate a bundling opportunity.

[8] relies on a given customer demand to drive the creation of service bundles. While this approach can be useful for situations where customer demand is well known and understood, poor performance can be expected when demand is hard to capture or anticipate. Furthermore, the economically desirable situation where customer demand is induced by a new service offering is not supported at all. Our proposed method explicitly targets the latter case by focusing on the creation of new and innovative service bundles. Therefore, customer demand is not utilized to reason about the suitability of potential bundles in this method. Instead, the driving source of

this method is a repository of services that are available for bundling. Depending on the given context, this repository might consist of the services of a single provider, a provider network or even contain all available services in a service ecosystem.

The bundling method created by [8] identified six distinct relationships that define dependencies between two services: core/enhancing, core/supporting, bundled, substitute, excluding, optional bundle. The (manual) evaluation of all services regarding these relationships is a prerequisite for the actual bundling process, as the feasibility of bundles is determined by the existing relationships. This evaluation is a time consuming task, one that becomes practically impossible to handle for a large set of services.

The proposed approach is therefore based on a service description which does not necessitate the step of explicating relationships between services. Instead, this method uses commonalities of attributes that indicate such a relationship. As long as services are consistently described and attributes relevant for this bundling approach are present, the proposed method can be employed. The following section will explain the term relationship as used in this method.

Leveraging Relationships between Services

[22] found that functionally complementary components in a bundle lead to high intentions to purchase compared to bundles in which no complementary components are present. The authors state that, "*as the relationship among the components increased from "not at all related" through "somewhat related" to "very related", intention to purchase also increased"*.

This method builds upon these findings and the conjecture that other commonalities or relationships between services can also indicate potentially useful bundles. For this method the term relationship is defined as *a connection, whose existence can be evaluated by a logic expression.* A relationship builds upon attributes from services' descriptions. Every relationship refers to previously specified attributes (e.g. location of a hotel, destination of a flight) and evaluates them using a given logic (e.g. distance between destination airport and location of the hotel). This evaluation can be realized ranging from simple value comparisons of single attributes to complex algorithms using multiple attributes. Figure 1 illustrates the coherence between the mentioned terms using UML.

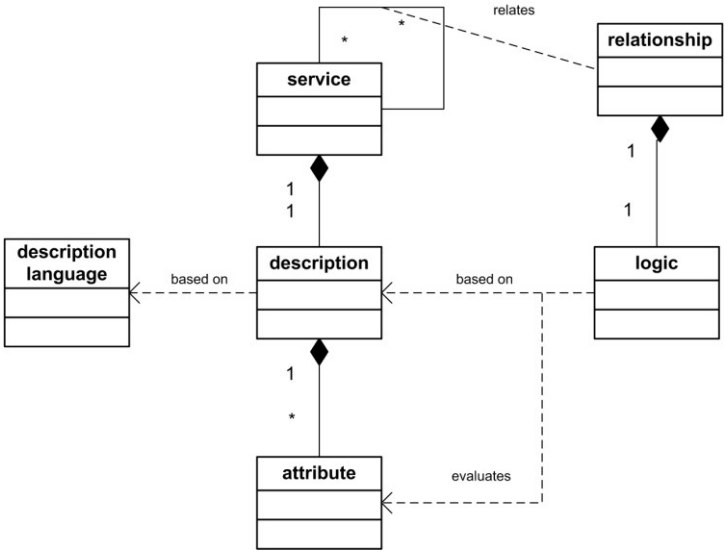

Figure 1: Concept of a Relationship

Relationships can display varying degrees of strength. For example, the distance between arrival airport and hotel determines the strength of this relationship. It depends on the concrete scenario and type of involved services as to whether a certain distance translates into a strong or weak relationship. Therefore the domain expert can configure the logic, where this is applicable. A first set of empirically derived relationships are presented in [10].

4 Detailing the Research Methodology

The overall research can be positioned in the area of Information Systems and Service Science research focusing on design science [23], as the service bundling method can be regarded as an innovative artifact to solve a contemporary problem. As a proof of concept, the method will also be supported by a software prototype, which represents another artifact. The research will be classified according to the design science approach with a strong action research flavor. Within this approach, different methods will be applied to satisfy the relevance and rigor criteria and align with the guidelines postulated by [24]. The research methodology can be divided into five stages, which will be shortly described in the following.

During the *preparation stage* a thorough literature review will be conducted to synthesize existing knowledge and position the proposed research within the overall body of knowledge. Thus, initial, preliminary answers will be found for research questions. Hereby, the literature review will provide initial answers regarding the nature and characteristics of services/SPM, the different types of bundling, the existing service description languages and service bundling/development

methodologies. These insights will be used to articulate a preliminary draft of a service bundling method.

The second stage, *analysis stage*, aims at deriving empirical insights into the foundations of bundling by conducting a content analysis of existing bundles and services. The scope of the analysis will include traditional bundles found in the business domain as well as novel, integrative bundles found in the IT domain. The outcome of this stage will be a set of factors that potentially contribute to the composition and in particular support reasoning about the feasibility of bundles. The content analysis will comprise the development of a questionnaire to identify initial relationships between services in a bundle that can be used to reason about the suitability of service bundles. Rigorous design science research must also take into considerations existing theories and frameworks. As part of this stage, existing theories will be analyzed to support a theory-driven design of the bundling method.

Once the related literature review and content/theory analysis have been conducted, it is envisaged to conduct at least one case study to complement the theoretical findings with empirical data about the current and desired situation as part of the *exploration stage*. The exploratory case studies are of an observable nature. The main objective is to find out about current SPM practices. As such we want to analyze the current way of describing services, practices of bundling services and portfolio management as well as requirements for a decision-support tool and method for service bundling as part of SPM. The requirements will be used at a later stage to design and validate the main artifact.

The fourth stage, *design stage*, is focused on designing the service bundling method (and its tool-support). Hence, this stage can be regarded as the core of design science research [23-25]. Once the final set of dimensions and requirements have been empirically identified, a gap analysis needs to be conducted to analyze the deficiencies of existing frameworks/methodologies and service description languages. The gap analysis will provide important insights of possible extensions of existing work as well as additional requirements for the service bundling method, which will be developed in this stage. The design phase of the artifact will be nourished by the results of the previous stages. Hence, insights gained through the literature review will be incorporated as well as insights gained through the case studies. Furthermore, empirical and theoretical insights gained in stage two as well as the requirements gathered in stage three will provide the basis for the specification of scope and features of the method and tool. The outcome of this stage is a tool-supported service bundling method that satisfies the requirements of the case study partners (relevance) and has been built upon existing knowledge and findings (rigor).

The *evaluation stage* focuses on evaluating the developed bundling method. According to the guidelines postulated by [24] and the requirements of design science by [25], an artifact needs to prove its utility and validity in a real world scenario. As the developed artifact of this study is a tool-supported service bundling method, a valid research method that aligns with the objectives of the study and the interest of the candidate is action research. Hereby, existing methods and expertise will be applied from the knowledge base to evaluate the tool-supported bundling method. The final outcome of this stage will be an evaluated tool-supported service bundling method.

5 Current Status

As pointed out previously, the first stage comprises activities to arrive at a synthesized view on the existing knowledge base related to services / SPM, different types of service bundling and a first draft of the bundling method. The results of this stage have been published in [4, 5] and [26] respectively.

In regard to stage two, we derived a first working set of generic relationships based on the proposed content analysis and subsequent empirical studies. Results have been synthesized and are published in [10]. An analysis of potentially applicable theories is still ongoing.

As pointed out in the last section, we aim at conducting at least one exploratory case study to gain empirical insights into SPM and bundling practices. As part of this stage, we were able to commence a comprehensive case study with a government authority. Hereby, it is envisaged to bundle services that can be accessed by potential consumers, namely public citizens, by utilizing the online channel. Hence, our research in that area touches upon current advances in the area of e-government as it aims at identifying ways to enhance the consumer satisfaction through the specification of service bundles that are presented on the government's webpage. The bundles comprise services that are offered by multiple departments to ease access and enhance the overall consumer satisfaction. By doing this, consumers no longer need to consult multiple departments and their respective websites as all services of potential interest are comprised within specific bundles. Furthermore, citizens do not need to know the internal structure of governments anymore in order to find their services of interest.

As part of this case study we will be able to accompany this specific government on its way to implement a customer-centric one-stop portal that comprises the presentation of service bundles. Currently we are observing market studies that aim at analyzing the potential benefits of offering services in bundles as part of a one-stop portal strategy. Furthermore, we are conducting several interviews with multiple other governments that already implemented such a one-stop portal based on service bundling activities in order to gain additional empirical insights. Finally, we will take part in all activities that are directly conducted by our case study partner in order to achieve their objectives successfully, such as conducting usability testing sessions with a representative sample of the targeted user group. This case study provides us with an unique opportunity to gather empirical insights into the bundling process and helps us to answer our research questions. First findings look very promising and will be in a more advanced stage in the near future.

Regarding stages four and five, we first need to successfully finish the other stages before we can actively target the activities in these stages.

6 Conclusion

This paper describes a research project proposing a novel approach to identifying service bundle candidates. Because of its potential to combine innovation with cost-effective re-use of existing services, we envision that service bundling will become as

important as new service development as, for example, can be seen in the growing attention for mash-ups. However, while the process of new service development has been extensively researched and conceptualized, the process of finding suitable service bundling candidates is still ill-defined.

The described research project proposes a method that facilitates the creation of bundles by providing organizations with systematic and practical guidelines. The method is a contribution to design science research in the field of Information Systems and Service Science. It represents an innovative artifact that extends the academic knowledge base related to service management. The developed method builds on service bundling concepts from both the marketing and the technological literature, thereby addressing the increased need for business-IT alignment. As such, it also is an example of a multi-disciplinary approach that builds on existing research in different areas and extends this research in new directions.

Based on the descriptions and explanations in the previous sections, multiple directions for further research can be identified. First, the "service bundling" task needs to be positioned as part of a management discipline. First insights suggest to position service bundling as a key task of service portfolio management, but further research needs to be conducted. Second, research in the area of service descriptions has to be conducted to develop a universal service description language that is applicable across industries and covers business as well as software services. Alternatively, extant service description languages need to be analyzed to determine in how far they accommodate the identified relationships and provide possibilities to be extended. Third, strategies and rationales of service bundling need to be analyzed further, to provide valuable insights for the internal and external validation of initially identified bundles.

Acknowledgements: This research was carried out as part of the activities of, and funded by, the Smart Services Cooperative Research Centre (CRC) through the Australian Government's CRC Programme (Department of Innovation, Industry, Science and Research).

References

1. Papazoglou, M.P.: Web Services: Principles and Technology. Prentice Hall, Edinburgh Gate, Harlow (2008)
2. Schroth, C.: The Service-Oriented Enterprise. Journal of Enterprise Architecture 3, 73--80 (2007)
3. Chesbrough, H., Spohrer, J.: A Research Manifesto for Services Science. Communications of the ACM 49, 35--40 (2006)
4. Rosemann, M., Fielt, E., Kohlborn, T., Korthaus, A.: Business Service Management In: Smart Services CRC White Paper Series pp. 1--14. Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia (2009)
5. Kohlborn, T., Fielt, E., Korthaus, A., Rosemann, M.: Towards a Service Portfolio Management Framework. In: Proceedings of the 20th Australasian Conference on Information Systems. Melbourne, Australia (2009)
6. Lawless, M.W.: Commodity Bundling for Competitive Advantage: Strategic Implications. Journal of Management Studies 28, 267--280 (1991)

7. Akkermans, H., Baida, Z., Gordijn, J., Peiia, N., Altuna, A., Laresgoiti, I.: Value Webs: Using Ontologies to Bundle Real-World Services. IEEE Int. Systems 19, 57--66 (2004)
8. Baida, Z.: Software-Aided Service Bundling. Intelligent Methods & Tools for Graphical Service Modeling In: Dutch Graduate School for Information and Knowledge Systems. Vrije Universiteit, Amsterdam (2006)
9. Gordijn, J., de Kinderen, S., Wieringa, R.: Value-Driven Service Matching. In: Proceedings of the 16th IEEE International Requirements Conference, pp. 67--70. IEEE Computer Society Washington, DC, USA (2008)
10. Kohlborn, T., Luebeck, C., Korthaus, A., Fielt, E., Rosemann, M., Riedl, C., Krcmar, H.: How Relationships Can Be Utilized for Service Bundling. In: Proceedings of the Americas Conference on Information Systems. Lima, Peru (2010)
11. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. 2 edn. Prentice Hall, NJ (2002)
12. Dreyfus, H.L.: What Computers Can't Do: The Limits of Artificial Intelligence. 2 edn. Harper and Row, New York (1979)
13. Negash, S., Gray, P.: Business Intelligence. In: Proceedings of the Americas Conference on Information Systems. pp. 3190--3199 (2003)
14. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proceedings of the ACM SIGMOD Conference on Management of Data. pp. 207--216 (1993)
15. Poole, D., Smyth, C., Sharma, R.: Semantic Science: Ontologies, Data and Probabilistic Theories. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., J., L.K., Lukasiewicz, T., Nickles, M., Pool, M. (eds.): Uncertainty Reasoning for the Semantic Web I. Lnai/Lncs, Vol. 5327/2008. pp. 26--40 Springer, Berlin, Heidelberg (2008)
16. Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5, 199--220 (1993)
17. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feier, C., Bussler, C., Fensel, D.: Web Service Modeling Ontology. Applied Ontology 1, 77--106 (2005)
18. Papadimitrou, C.H., Raghavan, P., Tamaki, H., Vempala, S.: Latent Semantic Indexing: A Probabilistic Analysis. Journal of Computer and System Sciences 61, 217--235 (2000)
19. Prokopp, C.: An Experimental Approach to Unsupervised Semantic Information Categorization by Selective Clustering and Tessellation of Associations. University of Queensland, Brisbane (2006)
20. Chu, H.: Information Representation and Retrieval in the Digital Age. Information Today, Medford, NJ (2007)
21. Riedl, C., Boehmann, T., Rosemann, M., Krcmar, H.: Quality Management in Service Ecosystems. Information Systems and E-Business Management 7, 199--221 (2009)
22. Herrmann, A., Huber, F., Coulter, R.H.: Product and Service Bundling Decisions and Their Effects on Purchase Intention. In: Fuerderer, R., Herrmann, A., Wuebker, G. (eds.): Optimal Bundling: Marketing Strategies for Improving Economic Performance. pp. 253--268 Springer (1999)
23. Hevner, A.R.: A Three Cycle View of Design Science Research. Scandinavian Journal of Information Systems 19, 87--92 (2007)
24. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design Science in Information Systems Research. MIS Quarterly 28, 75--105 (2004)
25. March, S.T., Smith, D.: Design and Natural Science Research on Information Technology. Decision Support Systems 15, 251--266 (1995)
26. Kohlborn, T., Luebeck, C., Korthaus, A., Fielt, E., Rosemann, M., Riedl, C., Krcmar, H.: Conceptualizing a Bottom-up Approach to Service Bundling. In: Proceedings of the 22nd International Conference on Advanced Information Systems Engineering (CAiSE'10). Hammamet, Tunisia (2010)

Evaluation and Improvement of Database Schemas: A transformation-based framework

Jonathan Lemaitre

Laboratory of Database Application Engineering - PReCISE research Center
Faculty of Computer Science, University of Namur
Rue Grandgagnage 21 - B-5000 Namur, Belgium
jle@info.fundp.ac.be
http://www.fundp.ac.be/precise

Abstract. Data schemas are primary artefacts for the development and maintenance of data intensive software systems. As for the application code, one way to improve the quality of the models is to ensure that they comply with best design practices. In this paper, we present a research on data schemas quality, in which the quality evaluation process is based on the identification of specific schema constructs and their comparison with best practices. We provide an overview of a framework based on the use of semantics-preserving transformations as a way to compare and suggest improvements for the most significant best design practices. The paper also summarize the contribution of the approach and of the current achievements.

Keywords: Database schema, schema evaluation, schema improvement, schema transformation.

1 Introduction

Quality has become one of the main topics in software engineering. Research and industrial communities acknowledge that behind maintainability, efficiency, etc., lies the satisfaction of the users and the financial impacts. The question has been studied through many works during the last three decades. In the 90's authors have already assessed the impact of poor quality and errors made during the modeling phase [1]. During the last few years quality of schemas became more and more important owing to the MDE approach that relies mainly on modeling and schema transformations.

Looking at data schema, one can observe different approaches to deal with quality. A first approach regroups the quality model and framework. Quality models are mainly composed of definitions of quality characteristics [2, 3] while in addition frameworks define high level views of the quality and methodologies [4–6]. Using quality models and frameworks, some authors propose metrics [7, 8]. Metrics give a numerical evaluation of (a specific aspect of) the quality. The metric approach deals with the evaluation of quality, most of the time, in a

affordable way, but it does not allow one to precisely identify the source of a poor quality score. Finally, the last approach regroups proposals, that study very specific problems such as, for example, the normalization [9] and the impact of particular construct [10, 11]. These proposals are generally based on validated and intuitive concepts. They also offer direct means to improve the quality of models. Combined with general modeling conventions, they can define a set of commonly agreed *best practices* that are to ensure specific requirements such as expressiveness, maintainability, evolutivity, performance, etc.

Both the metric approach and the study of specific problems have their advantages and limitations. In this paper, we will describe an ongoing research in which defects and best practices are considered as a possible basis for the schema quality evaluation and improvement. Our approach relies on the identification of schema structures that have been defined to express specific types of facts of the application domain. This analysis is used for evaluating the schema by comparing its structural content to a reference frame and the requirements of the schema context. The improvement activity will modify the schema structure, using semantics-preserving transformations, in order to increase its compliance with the context while preserving its semantics. Our work address the following questions:

- What can be the contribution of semantics-preserving transformations to the quality improvement of data models?
- What kind of problems/defects could be addressed this way?
- How to provide quality evaluation methods in order to assess the impact of transformations?

In section 2, we summarize the general concepts used in our research. Those are about schema abstraction level and paradigm and schema transformation. In section 3, we present our motivations and illustrate the type of quality problems we are studying. Section 4 presents the basis of a framework for the evaluation and improvement of schema quality. In the section 5, we compare our approach to related existing proposals. Finally, the section 6 describes the results achieved so far and gives directions about future works.

2 Background

In this section we briefly describe the main basic concepts used in our work. They are the abstraction levels and paradigms and the transformational approach. The interested reader is referred to reference [12] for a more detailed description of these.

A schema is a formal description of the information/data structures of a database, be it in construction or in use. It is positioned at a certain level of abstraction. Database engineering processes generally rely on a hierarchy of 3 abstraction levels, namely the conceptual, logical and physical levels. Such multi-level approach is currently called *Model-Driven Engineering*. A schema is

also expressed in a specification language, based on a definite paradigm. Entity-relationship (ER) with its many variants, UML class diagrams, relational, object-relational, XML, IMS, standard files and even *schema-less*, are some of them. The database community calls them *models* (e.g., the *relational model*), a term we will use in this paper. There is an agreement on which abstraction level a given paradigm best fits. For instance, the Entity-relationship model is as its best at the conceptual level while the object-relational model should be used at the logical level. Abstraction levels and paradigms define a two-dimension space in which an arbitrary schema can be located and evaluated.

Any process that consists in deriving artefacts from other artefacts relies on such techniques as renaming, translating, restructuring, replacing, refining and abstracting, which basically are transformations. Most database engineering processes can be formalized as chains of elementary models and data transformations that preserve some of their aspects. This approach is known as the transformational approach. In this paper, we will address the multiplicity of representations of a given concept by the use of specific transformations called semantics-preserving. Considering a transformation as a function, say g(c), defined on a set of constructs C, g is semantics-preserving or reversible iff there exists an inverse g', defined on C' such that, for each valid instance c of C, g(c) is a valid instance of C' and c = g'(g(c)).

3 Motivations

The quality of a schema can be seen through various aspects. One can deal with the visual quality [13], that relates to the graphical representation of models and properties of the schema objects (e.g. object position). Quality also encompass the syntactical correctness. The semantic quality refers to the correctness and completeness of the schema in regard to the application domain. Quality address problems such as the unsatisfiable constructs. These are syntactically valid constructs that cannot be instantiated because of the constraints it contains [14, 15]. One can also talk about quality in term of representation choice. In the previous section, we introduced the notion of abstraction level and paradigm. They defined a set of requirements on the schema, implying that some constructs may be incongruous for representing some elements of the application domain. Those constructs cannot be called errors, as they are syntactically and semantically correct. Instead, we use the term *defect*.

The figure 1 provides 2 examples of defects. The schema *(a)* contains an is-a relationship with one empty subtype. The use of the is-a relationship is not necessary, not to say unwise, if the subtype `FORMER-CUSTOMER` is not intended to evolve. Indeed, as `FORMER-CUSTOMER` has no role, nor attribute, it is said to be weakly specific and could be replaced by a simple boolean attribute of `CUSTOMER`. By doing such a transformation, we simplify a non-minimal schema. In the schema (b), one can observe an unusual construct coming from another abstraction level and paradigm (considering (b) to be an ER-like schema). In-

deed, the entity types, together with their relationship types, form a complex but valid expression of a single many-to-many relationship type. Such construct is very common in legacy IMS databases. Considering schema *(b)* at the conceptual level, all objects in the schema, considered separately, belong to the good paradigm but their combination forms a construct influenced by another abstraction level and paradigm.

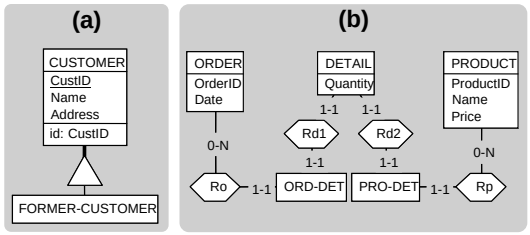

Fig. 1. Context-dependent defects

The transformational approach allow us to deal with the quality through the identification of defects, considering their possible existing alternatives and to improve schemas in order to make them compliant with best practices.

In the remaining of this section, we present defects identified for conceptual ER-like schema considering their understandability, which is one of the main requirement for conceptual schemas. The understandability refers to the efficiency with which a construct can express a type of fact of the application domain. For example, in order to represent a category of concepts A and its subcategories A_1 and A_2, the best solution will be to use an is-a relationship. It is the most expressive construct for such type of facts. However, alternatives exists and appear in schemas. A common alternative is the materialization of the is-a using one-to-one relationship types.

The identified defects have been regrouped into 5 categories. A full description can be found in [16]. Those categories are the non minimal, the insufficiently expressive, the abnormal, the irregular and the redundant constructs. For most categories, we will list the identified defects, but we can only detail some of them due to space restriction.

The **non minimal** constructs refer to the schema simplicity, which means that a type of fact should be represented as simple as possible. Attribute entity types are entity types that represent by *value* or *instance* attributes coming from another entity type. In order to simplify the schema, they can be represented in this other entity type. Empty and unique subtypes can be use to represent a simple property of the supertype, but should rather be expressed through a indicator in the supertype. Other non minimal defects are: N-ary relationship types with a [1-1] or [0-1] role; compound attributes with only one component; one-to-one relationship types with mandatory roles; weakly specified subtypes; split existence constraints.

Insufficiently expressive constructs relate to the expressivity of a schema. The expressivity requires that a construct express clearly and naturally its nature. Those defects regroup: relationship entity types; complex attributes; reference attributes; existence constraints containing roles and attributes; implicit is-a relationships. We talk about relationship entity type when a many-to-many relationship type is expressed with an entity type and 2 one-to-many relationships. Obviously, the many-to-many relationship is more expressive. A reference attribute is an implicit reference to another entity, that can be replaced by an explicit relationship.

Abnormal and **irregular** constructs decrease the foreseeable nature of a schema. Indeed, the user should not be surprise by the use of particular constructs. **Abnormal** constructs are: degenerated structures; entity types with no attribute, nor role; foreign constructs. A structure is degenerated if its composition is unjustified, e.g. a coexistence constraint with only one component or an is-a with a total constraint but only one subtype. Such defect should be solve by removing the unjustified element. Foreign constructs show a coloration coming from another paradigm and/or abstraction level (e.g. IMS,CODASYL), and may arise from a migration process. Those constructs will be replaced with the corresponding construct of the conceptual level.

Irregular constructs introduce a notion of uniformity between the constructs representing the same type of facts. In order to remove this type of defects, a choice of representation has to be done and applied to the whole schema.

Finally, the last category of defects concerns the **redundant** constructs. A schema should express a fact only once. Redundancies should be identified and removed in order to make the schema more understandable.

These defects can be removed by applying semantics-preserving transformations. Using the transformational approach allows us to apply similar reasoning with other abstraction levels, paradigms and quality characteristics. For example, at the logical level, the operational performance could be more interesting. The evolutivity of the schema may also be preferred at the conceptual level. To summary, the use of a construct C of a schema should be evaluated through three questions: Does C naturally belongs to this paradigm? Does C *feel comfortable* (so to speak) at this abstraction level? Does it best translate the intention of the designer?

4 Framework proposal

The goal we have chosen to reach in this research is to design a quality evaluation and improvement framework for data schemas. In particular, we expect (1) to augment global quality evaluation approaches with metrics based on semantically rich structural patterns and (2) to associate with each structural pattern correction transformations, in order to improve schema quality considering context requirements. Such transformations can be either suggested or automatically applied. Especially, we are evaluating the use of semantics-preserving transformations. The framework is based on the principle that these transformations

allow the production of alternative structures. Among the many possible structures, some of them, though correct, may not be considered best practices, while others may meet all requirements imposed by the context and therefore be considered as best practices. A more complete definition of the framework can be found in [12].

The concept of semantic equivalence class

In order to formalize our view of alternative structures, we defined the concept of equivalence class (EC). We consider K, the collection of all the constructs of the GER that are pertinent in some engineering processes and a set of transformations T. Let us also consider a construct C from K and all the equivalent constructs that can be derived through the reversible transformations of T. All these constructs, together with C, form an equivalence class called $ec(C)$. Since only reversible transformations have been applied, $\forall C' \in ec(C), ec(C') = ec(C)$. We now define the function $sec : K \rightarrow (K \times 2^K)$. $sec(C)$ associates to each construct in K its semantic equivalence class (sec), an equivalence class in which the specific element C has been tagged. C is the *intention* of this equivalence class. $sec(C)$ provides all the constructs a designer can introduce in a schema to express the semantics (the application domain fact type) of C, hence the name *semantic equivalence class* or *sec*.

Among a SEC, we consider a *best* structure, that is, the most suitable structure for expressing the modeling intention. Such structure can generally be considered as the best practice of the SEC in term of expressiveness. However, as discussed previously, depending on the context the use of this structure is not the best solution.

Context and SEC

In order to understand the use of the SEC, we need to define in more detail the concept of *context*. The *context* of a schema S is a set of requirements defined by the intended use of S. S has been designed for the abstraction level A, according to the paradigm P and to meet the design criterion D. We call *(A,P,D)* the context of S. Given a construct C that can appear in schema S, a scoring function is assigned to $sec(C)$ for a given context. As the SEC are defined independently of any model, we propose the concept of *projection* in order to take into account the model used in a particular context. The *projection* of a SEC for a model M provides the subset of all constructs of the SEC that comply with M.

Generation and representation of the structures

The application of transformations for generating the SEC should be considered carefully. The equivalence class of a construct C can be obtained by recursively applying the transformations of T until no new construct can be produced.

However, this naive approach can lead to a very large (and, depending on T, possibly infinite) set of constructs of which only a small subset would be of interest. Appropriate meta-rules are necessary to keep the process into reasonable limits. Considering the *is-a* pattern, one can adopt a *regularity of treatment* meta-rule according to which each sub-category of a given category must be expressed in the same way. For example, a construct obtained by applying the upward inheritance transformation to one sub-category and the materialization transformation to another one would be rejected. Another example: when an entity type EA results from the transformation of an attribute A, the attribute(s) of the latter cannot be further transformed through the same transformation (figure 2).

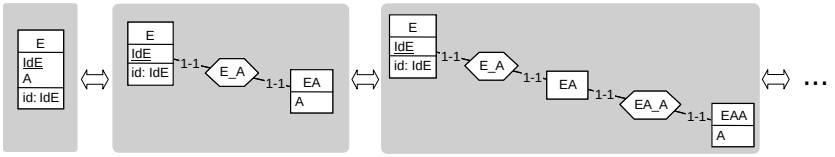

Fig. 2. Infinite transformation of an attribute.

Another important aspect of our framework is the use of generic definitions, or patterns, for representing the studied constructs. While the use of concrete examples, such as in the figure 1, for illustrating the considered defects is natural, it is completely unrealistic to attempt to list all of them without introducing a certain level of genericity.

5 Related works

The normalization process proposed by Codd [9] relies on the use of transformations in order to eliminate problematic functional dependencies. Compared with our framework, it deals with a *no redundancies* quality criterion. Burton and Weber [11] and Gemino and Wand [10] have studied particular constructs and their influence on specific qualities. Even though they did not explicitly refer to reversible transformations, our work seek to address similar problems. An important basis of our work is the proposal of Assenova and Johanesson [17]. They have considered the use of reversible transformations for enhancing the quality of conceptual data schemas. However, they associated quality *scores* directly to transformations, while we consider it to be related to the construct itself. Finally, Kurtev [18] used the concept of *transformation space* for dealing with schema quality. Such space represents a transformation by its initial and resulting structures and allows to link it with quality indicators. However, studied objects are atomic, while we consider semantically richer constructs.

6 Achievements and future works

So far, we have identified about 20 SEC, through a schema review process. All SEC represent some of the most common modeling intentions, that can be regrouped into 4 main categories: concepts, properties, relationships and constraints. Those categories are generally related to specific types of objects, e.g. entity types and tables for the concepts and attributes and columns for properties. The SEC regroups different constructs richer, in term of semantics, than simple objects. Besides the SEC, specific types of defects have also been identified and classified (e.g. non-minimal constructs, unexpressive constructs). The relation between SEC and those specific defects is currently studied.

Using SEC for enhancing the quality of schemas requires the ability to compare the quality between constructs. We have considered different approaches: the standard empirical studies, the use of metrics and the evaluation by experts. Among them, we choose to evaluate the quality of structures through the opinions of experts, which seemed to us to be a good compromise between the cost and the validity of the evaluation. Experts will be asked to assess each construct, independently of the schemas in which it appears.

The next step of this research is the complete quality evaluation of the SEC and their constructs. We should gather experts in order to obtain their quality rating. Such rating will allow us to produce *construct-based* metrics and will be the basis of the improvement process. The improvement method has to be defined in detail and carefully considering limits of our approach. For example, constructs belonging to different SEC could have common objects in a schema. Consequently, conflicts may appear between possible improvements.

An important goal of our research is to ensure the usability of the framework. This cannot be realized without a tool support for the identification of the constructs in a schema and the application of an improvement process. However, such tool has to be semi-automatic, due to limits of the approach. As the definitions of the SEC constructs is generic and represent structural properties, the same patterns may appear in different SEC. In such case, the identification of the modeling intention between the different possibilities has to be done manually by the analyst.

It remains to check the validity of the framework. Here, we wish to rely on teachers and students. (Last year) students form a realistic sample of designers of various skills, ranging from desperately inapt to experienced and ingenious. On the other hand, teachers are expected to be expert in evaluating the quality of medium size schemas. Therefore, comparing and aligning academic and automated evaluations allow the tuning of the evaluation framework. These validation and alignment processes are still under investigation.

References

1. Standish Group: Chaos: A recipe for success. Technical report, Standish Group International (1999)

2. ISO/IEC: ISO 9126-1:2001, Software engineering - Product quality, Part 1: Quality model. ISO/IEC (2001)
3. Davis, A., Overmyer, S., Jordan, K., Caruso, J., Dandashi, F., Dinh, A., Kincaid, G., Ledeboer, G., Reynolds, P., Sitaram, P., Ta, A., Theofanos, M.: Identifying and measuring quality in a software requirements specification. In: Proceedings of the First International Software Metrics Symposium. (1993) 141–152
4. Moody, D.L., Shanks, G.G.: Improving the quality of data models: empirical validation of a quality management framework. Inf. Syst. **28**(6) (2003) 619–650
5. Maes, A., Poels, G.: Evaluating quality of conceptual models based on user perceptions. In: International Conference on Conceptual Modeling - ER 2006, Tucson, Arizona (November 2006) 54–67
6. Krogstie, J.: Integrating the understanding of quality in requirements specification and conceptual modeling. SIGSOFT Softw. Eng. Notes **23**(1) (1998) 86–91
7. Manso, M.E., Genero, M., Piattini, M.: No-redundant metrics for uml class diagram structural complexity. In: CAiSE. Volume 2681 of Lecture Notes in Computer Science., Springer (2003) 127–142
8. Si-Said Cherfi, S., Akoka, J., Comyn-Wattiau, I.: Perceived vs. measured quality of conceptual schemas: An experimental comparison. In: ER (Tutorials, Posters, Panels & Industrial Contributions), Australian Computer Society (2007) 185–190
9. Codd, E.F.: Normalized data structure: A brief tutorial. In: SIGFIDET Workshop, ACM (1971) 1–17
10. Gemino, A., Wand, Y.: Complexity and clarity in conceptual modeling: comparison of mandatory and optional properties. Data Knowl. Eng. **55**(3) (2005) 301–326
11. Burton-Jones, A., Weber, R.: Understanding relationships with attributes in entity-relationship diagrams. In: ICIS '99: Proc. of the 20th international conference on Information Systems, Atlanta, GA, USA (1999) 214–228
12. Lemaitre, J., Hainaut, J.L.: Transformation-based framework for the evaluation and improvement of database schemas. In: Proc. of the 22st International Conference on Advanced Information Systems (CAISE'10, to appear). (2010)
13. Moody, D.: What makes a good diagram? improving the cognitive effectiveness of diagrams in is development. In Knapp, Magyar, eds.: Intl Conf on Information Systems Development, Budapest, Hungary, Springer (August 31-2 2006)
14. Boufares, F., Bennaceur, H.: Consistency problems in er-schemas for database systems. Inf. Sci. **163**(4) (2004) 263–274
15. Damm, F.M., Hansen, B.S., Bruun, H.: On type checking in vdm and related consistency issues. In: VDM '91: Proceedings of the 4th International Symposium of VDM Europe on Formal Software Development-Volume I, London, UK, Springer-Verlag (1991) 45–62
16. Hainaut, J.L.: Bases de donnees. Dunod (2009)
17. Assenova, P., Johannesson, P.: Improving quality in conceptual modelling by the use of schema transformations. In: ER '96: Proc. of the 15th International Conference on Conceptual Modeling, London, UK, Springer-Verlag (1996) 277–291
18. Kurtev, I.: Adaptability of model transformations. PhD thesis, University of Twente, Enschede (2005)

Acknowledgments This doctoral research is being led under the supervision of Pr. Jean-Luc Hainaut, Laboratory of Database Application Engineering - PReCISE Research Center, Faculty of Computer Science, University of Namur, Belgium.

Reverse Engineering User-Drawn Form-Based Interfaces for Interactive Database Conceptual Analysis

Ravi Ramdoyal

Laboratory of Database Application Engineering - PReCISE Research Center
Faculty of Computer Science, University of Namur, Belgium
rra@info.fundp.ac.be

Abstract. In this paper, we address the problem of eliciting, communicating and validating the static data requirements of a software engineering project, while improving the end-user involvement. For this purpose, given an environment for which electronic forms are a privileged way to exchange information and stakeholders familiar with form-based (computer) interaction, we propose to use form-based user-drawn interfaces as a two-way channel to interactively capture and validate static data requirements with end-users, by specializing and integrating standard techniques to help acquire data specifications from existing artifacts. Since the main principles of our approach are already presented in [1], we here focus on discussing two fundamental aspects of this research, namely the means to make end-users major stakeholders in the data requirements process, and the challenges facing the validation of such a transversal research.

Keywords: Information systems engineering, Requirements engineering, Database engineering, Human-computer interfaces reverse engineering.

1 Introduction

Requirements engineering is a key step in the realm of Software engineering, since its lays the ground work for further analysis, design and development. Within this process, Database engineering focuses on data modeling, where the static data requirements are typically expressed by means of a conceptual schema, which is an abstract view of the static objects of the application domain. Since long, conceptual schemas have proved to be difficult to validate by laymen, while traditional database requirements elicitation techniques, such as the analysis of corporate documents and interviews of stakeholders, usually do not actively and interactively involve end-users in the overall specification and development of the database. Still, the necessity to associate end-users of the future system with its specification and development steps has long been advocated [2]. In particular, the process of eliciting static data requirements should make end-users feel more involved and give them intuitive and expressive means to convey

their requirements to the analysts. Conversely, analysts should also be able to capture and validate these requirements by discussing them with the end-users.

In order to facilitate this communication, we present the tool-supported RAINBOW approach, which relies on reverse engineering user-drawn form-based interfaces to perform an interactive database conceptual analysis. This approach to elicit and validate database requirements is based on end-users involvement through interactive prototyping, and the adaptation of techniques coming from various fields of study. Since more details on our approach can be found in [1], the remainder of the paper is structured as follows. Section 2 briefly delineates the research context and related works. The main principles of our proposal are exposed in Section 3. In Section 4, we elaborate on the implications of an improved end-user involvement and the challenges facing the validation of our research. Finally, in Section 5, we discuss the merits and limitations of our proposal and anticipate future work.

2 Research Context

In our research, we focus on database conceptual modeling, through which user requirements are translated into a conceptual schema representing the application domain. The Entity-Relationship (ER) model has long been the most popular medium to express conceptual requirements [3], but this formalism often fails to act as an effective end-users communication medium because of its intrinsic complexity. Still, most users are quite able to deal with complex data structures that are expressed through more natural and intuitive layouts such as electronic forms [4].

This strong link existing between graphical interfaces and data models is usually exploited in forward engineering, by straightforwardly producing artifacts such as form-based interfaces from a conceptual schema, using transformational and generative techniques [5]. In particular, prototyping [6] often acts as a basis for interviews or group elicitation to provide early feedback. Conversely, a form contains data structures that can be seen as a particular *view* of a conceptual schema, which implies that Database reverse engineering [7] techniques can be applied to such interfaces to recover fragments of the conceptual schema.

Deriving requirements from prototype artifacts has a long tradition, but the number of studies on the subject is limited (especially recently) and several limitations must be underlined in most of them [1]. First of all, older studies do not intimately involve end-users in the database design process. More generally, the tools provided for the drawing of the interfaces are not dedicated to this purpose and/or not convenient for end-users. Secondly, the underlying form model of the interfaces must often be constructed by analyzing the layout of the form before its content. This is strongly related to the fact that the existing approaches also aim to create the final form-based interfaces of the future application. Regarding the coherence of these interfaces, it is assumed that the labels are used consistently through out the different forms, and little care is given to possible lexical variation (paronymy, feminine, plural, spelling, mistakes, etc.) and on-

tological ambiguity (polysemy, homography, synonymy). The use of examples (either through static statements or dynamic interaction) is not systematically used to elicit constraints and dependencies. And last but not least, these approaches do not use the form-based interfaces as a means for the analysts to validate the underlying integrated data model.

3 Proposal

Within the requirements engineering phase of a software engineering project, our research therefore addresses the combination of reverse engineering techniques with user-based prototyping in order to interactively involve end-users in the conceptual modeling of the application domain. More precisely, given an environment for which electronic forms are a privileged way to exchange information and stakeholders are familiar with form-based (computer) interaction, we propose to use form-based user-drawn interfaces as a two-way channel to interactively capture and validate static data requirements with end-users, in order to alleviate understandability limitations of the ER model.

To succeed, we need to overcome several challenges inherent to the involved fields and their combination. Regarding Database engineering, we notably need to clarify the terminology, elicit constraints and dependencies, handle schema integration and generate applicative components. Regarding Database reverse engineering, we must handle the extraction of data models from the form-based interfaces, and since we want to make the Prototyping interactive, we need to enable the expression of concepts through form-based interfaces and the testing of generated components. Finally, integrating these fields into an approach involving end-users implies managing this user implication and tailoring the techniques.

To handle these challenges and answer the concerns raised in Section 2, our RAINBOW approach is formalized into seven steps involving end-users in a simple and interactive fashion, using interfaces as a specification language rather than legacy artifacts (as in traditional Reverse engineering), and providing the analysts with semi-automatic tools. Let us give an overview of these steps, for which more details can be found in [1]: (1) *Represent*: After preliminary discussions and appropriate training, the end-users are invited to draw a set of form-based interfaces to perform usual tasks of their application domain. Such interfaces are typically entry forms to capture data on, say, a new customer or a new product. A dedicated drawing tool intentionally provides them with a constrained layout mechanism and a limited set of primitive widgets (namely *interfaces, group boxes, tables, input fields, selection fields* and *action buttons*). (2) *Adapt*: Once the interfaces are drawn, mapping rules are automatically applied to extract the underlying data models of the interfaces. (3) *Investigate*: The end-users and the analyst then jointly cross-analyze the interfaces to arbitrate the possible labeling ambiguities (lexically or ontologically similar labels) and structural similarities (containers owning widgets with the same labels) that are automatically identified in the interfaces and their underlying data models.

(4) *Nurture*: Using the forms they drew, the end-users then provide a set of positive and negative data samples, from which induction techniques allow to suggest possible constraints and dependencies that also need to be arbitrated. (5) *Bind*: The validated redundancies, constraints and dependencies are processed to perform an interactive integration of the key concepts elicited through the previous steps, hence leading to an integrated conceptual schema. (6) *Objectify*: A lightweight prototype application is generated from the integrated conceptual schema. It comprises a simple data manager that uses the interfaces drawn by the end-users and allows them to manipulate the concepts that have been expressed, typically to inspect, create, modify and remove data. (7) *Wander*: Finally, the end-users are invited to "play" with the prototype in order to ultimately validate the requirements, or identify remaining flaws.

In our doctoral research, we mainly focus on the five first steps, since the generation of the components is straightforward and the manipulation of a reactive prototype mainly adds another level of validation.

4 Discussion

In this section, we discuss the design and validation of our approach. More specifically, we ponder how the research context and the need for interactivity influenced the design of our approach, especially the tailoring or existing techniques, and we envision the obstacles threatening its assessment.

4.1 How to make end-users major stakeholders of the data requirements process?

The RAINBOW approach relies on the principles of the ReQuest framework [8, 9], which provides a complete methodology and a set of tools to deal with the analysis, development and maintenance of web applications. ReQuest deals with data modeling and the dynamic aspects of the future application, and proved that it is possible to efficiently and swiftly involve end-users in the definition of their needs. However, most laymen end-users were challenged by the task of designing dynamic and rich front-end interfaces supporting the business logic of their future application. Here, we therefore decided to focus specifically on improving the static data requirements process, leading the interfaces to appear as a means rather than an end product.

In particular, we wanted form-based interfaces to serve as a basis for discussion and joint development, and developed a tool to support this approach. To make the development of the interfaces more accessible and focus the drawing on the substance rather than (ironically) the form, we restricted the available graphical elements to the most commonly used ones and limited the layout of forms as a vertical sequence of elements [1], which also simplifies the mapping rules between the form model and the ER model. During the drawing, end-users must at least provide the label and cardinality of these elements, while advanced users may also provide integrity and existence constraints (which are normally

addressed during the following steps). The interfaces being drawn by non experts and possibly multiple end-users increases the possibility of inconsistencies among the labels used. In order to standardize the vocabulary from the start, we include a term analyzer that suggests alternative labels on-the-fly for new elements, using *String Metrics* [10] and the lexical reference system WordNet [11]. The same term analyzer is used during the *Investigate* phase to clarify any remaining ambiguity.

We mentioned in [1] the *structural redundancy* issue that we deal with during that same phase, and how our context led us to prefer a simple comparison algorithm to existing *frequent embedded subtrees mining* algorithms for rooted unordered trees. This case is very representative of the choices we made during the design of our approach. Take for instance the *Nurture* phase, during which we need to elicit integrity constraints (identifiers, cardinalities, value types and sizes, domain values, ...), existence constraints among optional fields (coexistence, at least one, exactly one, at most one, ...), and functional dependencies (when the values of certain fields may determine the values of other fields). Such constraints and dependencies can be discovered by analyzing existing data samples. Several techniques deal with this issue (candidate generate-and-test, minimal cover, ...), but they rely on large preexisting data sets. In our case, there is possibly no available data sample or the re-encoding would be too expensive, and it is anyways unrealistic to ask end-users to willingly provide numerous data samples.

These observations naturally called for new ways to discover and suggest constraints and dependencies on-the-fly, based on the incremental input of data samples by the end-users. Regarding the constraints, inductions can be made on these data samples to make suggestions on the value types (if all the instances of a given field are filled with numbers, this could suggest that the field is numeric or textual), the cardinalities (if all the instances of a given optional field are not empty, this could suggest that the field is actually mandatory), the existence constraints (when two optional fields are always filled at the same time, they could be coexistent), and so on...

As for functional dependencies (FDs), the ideal process should lead us to build a set of data samples and dependencies so that each entity type of the underlying conceptual schema becomes an Armstrong relation (i.e. a relation that satisfies each FD implied by a given set of FDs, but no FD that is not implied by that set). Reaching such a state is obviously not trivial, but we can try to near it progressively narrowing the FDs. We start by calculating the "strongest" valid candidates FDs for each entity type (i.e. each mandatory simple attribute of an entity type may determine the others), and maintain them until a data sample proves them wrong. Whenever a new data sample jeopardizes a functional dependency, the FD is discarded and valid alternative "lesser" FDs are recursively generated. To add interactivity, end-users can enforce or discard FDs by adding valid data samples for the entity type, arbitrating problematic data samples for a given FD (automatically generated from previously provided valid data samples), or even by directly enforcing or discarding FDs. This process should be repeated until there are only discarded and/or enforced FDs left for

each entity type. Possible identifiers should also be validated, knowing that an enforced FD $f : X \rightarrow Y$ may induce a identifier for the entity type E iff $X \cup Y = Attributes(E)$.

We consequently developed the RAINBOW tool kit, which is a user-oriented development environment, intended to assist end-users and analysts during the five first steps of our approach. The tool kit interacts with the repository of DB-Main, a database engineering CASE tool [12] providing all the necessary functionalities to support a complete database design process, as well as transformation tools and Database reverse engineering support. The interaction between these tools allows one to cover the whole database engineering process from both the end-user and the analyst perspectives.

4.2 How to validate the RAINBOW approach?

One of the most critical aspect of this research concerns its validation. The transversal nature of our approach, as well as the interdependence between the methodology and the tool support, naturally lead to the following research questions: (1) Does the RAINBOW approach and tool support help end-users and analysts to communicate (i.e. express, capture, validate) static data requirements to each other? (2) What is the quality of the conceptual schema produced using the RAINBOW approach?

The first question raises methodological (strategic design decisions), practical (potency and usability of the tool-support, added value for stakeholders) and sociological (end-user/analyst communication, empowerment, objectivity) issues, while the second question addresses the intrinsically complex predicament of quality assessment. Such problems are not easy to experiment, measure and validate, especially given the inherent difficulty of evaluating methodologies for the development of large systems : valuable learnings would require comparing our approach to existing ones, based on multiple experimentations led on numerous and different case studies over an extensive time span, which is not feasible at our level.

However, one of the contributions of our research is instead to define an experimentation canvas, based on preliminary studies that could in turn lead to a more realistic experimentation endeavor. The main idea to achieve this objective is to observe real-life implementations of our approach (using the *Participant-Observer* principles), then analyze the resulting conceptual schemas (using the *Brainstorming/Focus group* principles). Two independent studies have therefore been led, based on real-life issues concerning the two carefully chosen end-user participants EU1 and EU2. For each preliminary study, a pair of observers (including a main observer MO and a different assistant observer in each case) have therefore observed the interaction of one of the end-users with an analyst DB1 (the same in each case), jointly designing the conceptual schema of their dedicated project using the RAINBOW methodology and tool kit. Then, each resulting conceptual schema was discussed by three database analysts (DB1, DB2 and MO) to determine their qualities and flaws, as well as the delta between the "automatically" produced schemas and their "corrected" version.

Before starting the observation, EU1, EU2 and DB1 received a short training on the tool support and methodology based on screencast tutorials, and a session of questions/answers. The process was organized in four sequential sessions, each focusing on a peculiar aspect of the RAINBOW approach:

1. *Drawing the forms* (Represent): first of all, the end-users drew and edited the forms necessary to accomplish the tasks of their application project with the help of the analyst. They were asked to focus on the terminology and data aspect of this application, that is, the consistency of the labels and the specification of the widgets they needed, typically to input data.
2. *Analyzing the terminology of the forms* (Investigate): (1) the end-user and the analyst first analyzed the terminology of all the form elements to clarify any remaining ambiguities; (2) then, they analyzed the terminology of the containers to explain the relations existing between them. Whenever necessary, the pair went back and edited their forms.
3. *Providing examples and constraints* (Nurture): for each form, the pair first provided data samples then examined the technical constraints, the existence constraints, the functional dependencies and the possible identifiers associated with the form and its elements. Whenever necessary, the pair went back and replayed the previous steps.
4. *Finalizing the project* (Bind): from the previous steps, a set of "high level" concepts were materialized. For each of these concepts, the pair arbitrated the properties that were to be associated with the concept, then examined the associated technical constrains, the existence constraints, the functional dependencies and the possible identifiers. Whenever necessary, the pair went back and replayed the previous steps.

The observations and results of these preliminary studies must still be analyzed in depth, however they already raised several open questions, for instance regarding the drawing phase. During that step, the end-users naturally gave the commands to the analyst and were initially reluctant to manipulate the RAINBOW tool kit. On the other hand, the analyst did not feel very helpful or required for the process when he was not in charge of the drawing, though the end-users felt their presence reassuring. Who should therefore be drawing and who should be assisting? Is the drawing really a job for the end-user? What about the analyst's involvement and gratification?

We wanted to lead the end-user to focus on the content of the forms rather than their appearance, and subsequently chose an adaptative rendering for the widgets. For instance, selection widgets automatically switches from radio buttons to checkboxes or a selectable list according to the number of options and the cardinality of the field. However, this behavior surprised the end-users, and more generally they would have enjoyed at least a minimum of customization for the rendering of the widgets. Though the forms could be rendered afterwards in more stylish fashions (e.g. with HTML and CSS), could aesthetical considerations lead to a "bad" modeling, just because the end-users want the forms to be prettier? Can the analyst convince them that "it is ok even if it is ugly", and can the end-users really agree on that?

Likewise, the available widgets are restricted to forms, fieldsets, tables, inputs, selections and action buttons. For the two studies, these widgets seemed sufficient, although the composition sometimes called for creativity. We also observed that the end-users often drew single forms to collect multiple informations instead of drawing smaller, simpler forms (i.e. breaking the problems into smaller sub problems). Do the available widgets hence lead to the drawing of single oversized forms?

As previously mentioned, widgets have a cardinality specifying how many values could and should at least and at most be provided. We observed that the end-users often specified widgets as "mandatory", even if they sometimes acknowledged that it would not really be problematic if the given fields were not filled. Could the end-users therefore abusively use this type of cardinality while it is not really necessary? Do they understand the difference between a paper form, which can be submitted even if it is incorrect, and an electronic form which offers immediate acceptance or rejection?

As we can see, the drawing behavior of the end-users would make for an interesting research topic by itself, and so would the response of analysts to such an approach. These preliminary studies therefore highlight several sensible phenomenons that would require special attention on a larger scale experimentation. Besides, it would also be interesting to study the evolution aspects of our approach. While we already consider the possibility to "loop" during the steps of our approach as long as we are in the conceptual design, what would be the situation if we needed to edit a working database produced using our approach?

5 Conclusion

This paper presented a comprehensive interactive approach to bridge the gap between end-users and analysts during the conceptual modeling phase of database engineering. This approach supports the elicitation and validation of static data requirements with end-users, while overcoming several limitations of existing prototyping methods. It relies on the expressiveness and understandability of form-based user interfaces, used jointly with tailored Reverse engineering techniques to acquire data specifications from existing artifacts. Although our approach addresses a significant subset of these requirements, it does not cover all of its aspects, and therefore does not replace more traditional task and information analysis approaches, but rather complements them.

To get a better perspective on this research, we also discussed how the research context and the need for user involvement and interactivity influenced the design of our approach, especially the tailoring and combination of the existing techniques, as well as the inclusion of supportive tools.

Finally, we addressed the intrinsic difficulty to validate our transversal approach, while exposing a series of open questions raised by our ongoing preliminary studies. From the observations of these real-life implementations of our approach, we will define a set of guidelines for an experimentation canvas that could set the basis for a wider evaluation and an improved use of this approach.

Acknowledgments This doctoral research is being led under the supervision of Pr. Jean-Luc Hainaut, Laboratory of Database Application Engineering - PReCISE Research Center, Faculty of Computer Science, University of Namur, Belgium.

References

1. Ramdoyal, R., Cleve, A., Hainaut, J.L.: Reverse engineering user interfaces for interactive database conceptual analysis. In: Proc. CAISE'10 (to appear). (2010)
2. Rosson, M.B., Carroll, J.M.: Usability Engineering: Scenario-Based Development of Human-Computer Interaction. Morgan Kaufmann (October 2001)
3. Shoval, P., Shiran, S.: Entity-relationship and object-oriented data modeling-an experimental comparison of design quality. Data Knowl. Eng. **21**(3) (1997) 297–315
4. Terwilliger, J.F., Delcambre, L.M.L., Logan, J.: The user interface is the conceptual model. In: Proc. ER'06. Volume 4215 of LNCS., Springer (2006) 424–436
5. Hainaut, J.L.: The transformational approach to database engineering. In Lämmel, R., Saraiva, J., Visser, J., eds.: Generative and Transformational Techniques in Software Engineering. Volume 4143 of LNCS., Springer (2006) 95–143
6. Davis, A.M.: Operational prototyping: A new development approach. IEEE Softw. **9**(5) (1992) 70–78
7. Hall, P.A.V.: Software Reuse and Reverse Engineering in Practice. Chapman & Hall, Ltd. (1992)
8. Brogneaux, A.F., Ramdoyal, R., Vilz, J., Hainaut, J.L.: Deriving user-requirements from human-computer interfaces. In: Proc. IASTED'05. (2005) 77–82
9. Vilz, J., Brogneaux, A.F., Ramdoyal, R., Englebert, V., Hainaut, J.L.: Data conceptualisation for web-based data-centred application design. In: Proc. CAISE'06. LNCS, Springer (2006) 205–219
10. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A comparison of string distance metrics for name-matching tasks. In: Proc. IJCAI IIWeb Workshop. (2003) 73–78
11. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998)
12. DB-Main: The official website `http://www.db-main.be`.