The 6th SIKS Conference on

Enterprise Information Systems 2011

Delft, The Netherlands, October 2011

Proceedings

Virginia Dignum
Jan Hidders
Sietse Overbeek (Eds.)

TU Delft Delft University of Technology

Challenge the future

Volume Editors:

Virginia Dignum
Delft University of Technology
PO Box 5, 2600 AA Delft, The Netherlands
E-mail: m.v.dignum@tudelft.nl

Jan Hidders
Delft University of Technology
PO Box 5, 2600 AA Delft, The Netherlands
E-mail: a.j.h.hidders@tudelft.nl

Sietse Overbeek
Delft University of Technology
PO Box 5, 2600 AA Delft, The Netherlands
E-mail: s.j.overbeek@tudelft.nl

The papers and extended abstracts in this book comprise the proceedings of the 6th Conference on Enterprise Information Systems (EIS 2011), held on October 31st 2011 at the Delft University of Technology (TUDelft).

EIS 2011 is organized by SIKS (School for Information and Knowledge Systems) in cooperation with NAF (Nederlands Architectuur Forum). The conference offers a unique opportunity for research groups from both the Computer Science-side and the Management-side to report on research, meet and interact. We also welcome practitioners with an interest in research and innovation, as well as doctoral students in the early stages of their careers.

These proceedings also appear online at http://ceur-ws.org/

SIKS

http://www.cs.uu.nl/siks/

NAF

http://www.naf.nl

Proceedings of the

6th SIKS Conference on
Enterprise Information Systems 2011

Delft, The Netherlands,
31st October 2011

Virginia Dignum
Jan Hidders
Sietse Overbeek (Eds.)

Preface

In follow-up to the five previous conferences, which were held in Utrecht, Groningen, Tilburg, Ravenstein and Eindhoven, the sixth edition of the SIKS Conference on Enterprise Information Systems (EIS 2011) is held in Delft this year. The purpose of the conference series is to bring together Dutch and Belgian researchers interested in the advances in and the business applications of information systems. Against that backdrop, we are happy to have received contributions from almost all of the Dutch and Belgian research groups active in this field.

Overall, the program consists of 14 presentations of which three concern totally new work. Each of these was duly reviewed by at least three members of the program committee. The remaining 11 presentations relate to work already being published in a high-quality outlet and considered highly attractive to bring under the attention of the Dutch/Belgian EIS community.

On top of these presentations, we are very happy with the incorporation of two keynote presentations in the program. The opening keynote is to be given by Niek Wijngaards, senior researcher and program manager at Thales Nederland. The closing keynote will be provided by Marc Lankhorst who is Principal Researcher at Novay.

At this occasion, we wish to express first and foremost our gratitude to the members of the EIS community who have fulfilled roles in the program committee for this conference. Their valuable feedback has helped the presenters to further improve their work. We also wish to thank SIKS, NAF and all involved local staff at Eindhoven University of Technology for their support in organizing this event.

It is our hope that the conference will stimulate discussions in our community, foster existing collaborations and lead to new ones. But most important of all, we hope that you will enjoy the conference day.

October 2011

Virginia Dignum
Jan Hidders
Sietse Overbeek

Conference Organization

General Chairs:

Geert-Jan Houben
Yao-Hua Tan

Organizing Chairs:

Virginia Dignum
Jan Hidders
Sietse Overbeek

Program Committee:

Wil van der Aalst,	TU/e (Eindhoven)
Hamideh Asarmanesh,	UvA (Amsterdam)
Jan van den Berg,	TUD (Delft)
Patrick van Bommel,	RU (Nijmegen)
Harry Bouwman,	TUD (Delft)
Bert de Brock,	RUG (Groningen)
Hennie Daniels,	EUR (Rotterdam)
Flavius Frasincar,	EUR (Rotterdam)
Paul Grefen,	TU/e (Eindhoven)
Jaap Gordijn,	VU (Amsterdam)
Frank Harmsen,	UM (Maastricht)
Laura Hollink,	TUD (Delft)
Stijn Hoppenbrouwers,	RU (Nijmegen)
Gert-Jan Hofstede,	WUR (Wageningen)
Slinger Jansen,	UU (Utrecht)
Marijn Janssen,	TUD (Delft)
Patricia Lago,	VU (Amsterdam)
Marc Lankhorst,	Novay (Enschede)
Aldo de Moor,	CommunitySense
Michael Petit,	FUNDP (Namur)
Geert Poels,	KUL (Leuven)
Erik Proper,	CPR Henri Tudor (Luxembourg)
Hajo Reijers,	TU/e (Eindhoven)
Monique Snoeck,	KUL (Leuven)
Philippe Thiran,	FUNDP (Namur)
Jan Vanthienen,	KUL (Leuven)
Jan Verelst,	UA (Antwerpen)
Niek Wijngaards,	D-CIS (lab Delft)

Program

	Room A (Ampere)	Room B (Boole)
09:30	*Reception with coffee & tea*	
09:50	**Opening**	
10:00	**Keynote:** Niek Wijngaards *"Actor-Agent Communities: supporting inter-organisational cooperation"*	
11:00	*Morning break*	
11:30	**Session:** Enterprise Engineering	**Session:** Service Science
	"The Quest for Know-How, Know-Why, Know-What and Know-Who: Using KAOS for Enterprise Modelling", **Maxime Bernaert and Geert Poels.**	*"Agile Process for Integrated Service Delivery"*, **Marjana Shammi, Sietse Overbeek, Robert Verburg, Marijn Janssen and Yao-Hua Tan**.
	"Towards an Investigation of the Conceptual Landscape of Enterprise Architecture", **Dirk Van Der Linden, Stijn Hoppenbrouwers, Alina Lartseva and Erik Proper**	*"Agile Service Development: a Rule-Based Method Engineering Approach"*, **Stijn Hoppenbrouwers, Martijn Zoet, Johan Versendaal and Inge Van De Weerd**.
	"Issues and Challenges in Dynamic Systems Design and Engineering – A Value-Oriented Approach", **Joao Pombinho and Jose Tribolet**	*"Towards a Service System Ontology for Service Science"*, **Elisah Lemey and Geert Poels.**
12:30	*Lunch, served in EWI restaurant on the 8th floor*	
14:00	**Session:** Business Process Modeling	**Session:** Collaborative Modeling
	"Business Process Simulation for Management Consultants: A DEVS-Based Simplified Business Process Modelling Library", **Igor Rust, Deniz Cetinkaya, Mamadou Seck and Ivo Wenzler**.	*"Assessing Collaborative Modeling Quality Based on Modeling Artifacts"*, **Denis Ssebuggwawo, Stijn Hoppenbrouwers and Erik Proper.**
	"Merging Computer Log Files for Process Mining: an Artificial Immune System Technique", **Jan Claes and Geert Poels.**	*"A Context-aware Inter-organizational Collaboration Model Applied to International Trade"*, **Jie Jiang, Virginia Dignum, Yao-Hua Tan and Sietse Overbeek**.
	"Questioning the Design of Business Process Maturity Models", **Amy Van Looy, Manu De Backer and Geert Poels.**	*"Applying Soft Systems Methodology in Enterprise Architecture Creation Workshops"*, **Agnes Nakakawa, Patrick Van Bommel and Erik Proper.**
	"Business Service Modeling for the Service-Oriented Enterprise", **Jeewanie Jayasinghe Arachchige, Hans Weigand and Manfred Jeusfeld.**	*"Change your lifestyle or your game is over: The design of a serious game for Diabetes"*, **Harmen Nauta and Ton Spil**
15:20	*Afternoon break*	
15:50	**Keynote:** Marc Lankhorst, *"ArchiMate: Past, Present and Future"*	
16:50	*Closure and drinks in /pub in the EWI building basement*	

Table of Contents

Session 4: Collaborative Modeling

Morning Keynote

Title: *Actor-Agent Communities: supporting inter-organisational cooperation*

Speaker: Niek Wijngaards

Abstract

Actor-Agent Communities are social-technical systems including human and artificial entities. AACs have the capability of contributing to solving non-standard, real-life complex problems (such as decision making under critical circumstances, dynamic situational contexts, and distributed decision points), where satisfactory solutions can only be obtained through a genuine fusion between the creative thinking of humans and the processing speed, accuracy and endurance of computers. One of the interesting phenomena is that AACs can span across organisational boundaries, enabling cooperation and sharing of information while also retaining organisational autonomy and policy-making.

The keynote addresses actor-agent communities and in particular their application to foster secure, distributed multi-organisational information sharing, cooperation and context-specific information dissemination. The ideas and concepts are briefly illustrated by means of the recently started MIA-Veiligheid project "SlimVerbinden".

About the Speaker

Dr. Niek Wijngaards is senior researcher and program manager at Thales Nederland B.V. He studied Computer Science at the Vrije Universiteit Amsterdam where he specialised in artificial intelligence. He received his PhD in 1999 on the topic of self-modifying agent systems using a re-design process. In 1998 he worked for a year at the University of Canada as a Postdoctoral-Fellow, after which he became an assistant professor at the Artificial Intelligence group at the VUA. From 2000 to 2004 he was an assistant professor at the Intelligent Interactive Distributed Systems group at the VUA headed by prof.dr. Frances Brazier. During these years his research involved design processes to support large-scale heterogeneous adaptive multi-agent systems and analysis of legal implications of agent technology. Since October 2004 he works for Thales Research & Technology Netherlands, part of Thales Nederland BV, and is fully employed at D-CIS Lab, see http://www.d-cis.nl . Niek has a joint role as senior researcher and program manager, being involved in research on actor-agent teams as well as their practical applications at e.g. the Dutch Police organisation and the Dutch National Railroad. Niek participates in both international (FP7) projects such as DIADEM and BRIDGE, as well as national projects, including MIA-Veiligheid Slim Verbinden. Since 2011 he holds the position of Visiting Senior Research Fellow at the Leeds University Business School. His current research addresses cross-organisational cooperation using actor-agent communities and scenario-based multi-criteria decision analysis, see http://publications.decis.nl .

Afternoon Keynote

Title: *ArchiMate: Past, Present and Future*

Speaker: Marc Lankhorst

Abstract

In current business practice, an integrated approach to business and IT is indispensable. Take for example a company that needs to assess the impact of introducing a new product in its portfolio. This may require defining additional business processes, hiring extra personnel, changing the supporting applications, and augmenting the technological infrastructure to support the additional load of these applications. Perhaps this may even require a change of the organizational structure.

Enterprise architecture is an important instrument to address this company-wide integrated approach to development and change. It is a coherent whole of principles, methods and models that are used in the design and realization of the enterprise's organizational structure, business processes, information systems, and infrastructure. To create such an integrated perspective on enterprise architecture, one needs both a description technique for these architectures, and a method for architectural design in which this technique is employed. The ArchiMate® language was developed by Novay and partners from industry and academia to provide such a description technique. In 2008, has been transferred to The Open Group, which has also developed the main open method for architecture development, TOGAF®.

These two standards nicely complement each other. However, the scope of TOGAF is wider than that of ArchiMate 1.0. This has lead to the development of a second version of the language, extending it with concepts for modelling e.g. goals, requirements, projects and transitions. This will become the official ArchiMate 2.0 standard in the very near future. This talk will focus on the past, present and future of ArchiMate, the elements of the language, practical experiences, and the relationship with TOGAF.

About the Speaker

Marc Lankhorst is Principal Researcher at Novay, where he is responsible for the activities on enterprise, business and IT architecture. His expertise and interests ranges from enterprise architecture and business process management to service orientation and agile software development. In the past, Marc has managed the ArchiMate project, a major cooperation between several partners from government, industry and academia, which developed the international ArchiMate standard for enterprise architecture description and resulted in the book Enterprise Architecture at Work (currently in its second edition; Springer, 2009). He has published over 100 scientific and business papers in journals, magazines, books and conference proceedings.

In recent years, much of his work has been on architecture issues in service delivery and cooperation between organizations, e.g. on integrated e-services from government and industry, multichannel management, and interoperability.

Marc is a board member of the Netherlands Architecture Forum (NAF) member of the ArchiMate Forum of The Open Group, one of the editors of Via Nova Architectura, organizer of several workshops, conferences and seminars, and TOGAF9 certified enterprise architect.

Marc holds an MSc from the University of Twente (1991) and a PhD from the University of Groningen (1996).

Session 1: Enterprise Engineering

The Quest for Know-How, Know-Why, Know-What and Know-Who: Using KAOS for Enterprise Modelling

Maxime Bernaert and 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,Geert.Poels}@UGent.be

Abstract. While the field of information systems engineering is largely focussed on developing methods for complex problems and larger enterprises, less is done to specifically address the needs of smaller organizations like small and medium sized enterprises (SMEs), although they are important drivers of economy. These needs include a better understanding of the processes (know-how), why things are done (know-why), what concepts are used (know-what) and who is responsible (know-who). In this paper, the KAOS approach is evaluated as not only useful for developing software projects, but with the potential to be used for developing a business architecture or enterprise model. An example of KAOS is given, by way of illustration, and KAOS was applied in a case study by an SME's CEO, which resulted in a set of questions for further research.

Keywords: Requirements Engineering, Business Process Management, Small and Medium Sized Enterprises, Goal Modelling, KAOS, Business Architecture, Enterprise Architecture, Enterprise Modelling

This paper has been published as: Bernaert, M. and Poels, G. (2011): The Quest for Know-How, Know-Why, Know-What and Know-Who: Using KAOS for Enterprise Modelling. In: Advanced Information Systems Engineering Workshops, C. Salinesi and O. Pastor, LNBIP 83, pp. 29-40, Springer-Verlag Berlin Heidelberg (2011).

Smaller organizations, like small and medium sized enterprises (SMEs), require proper systems to fulfil their information and automation needs, but their first concerns are organizational issues, with IT as a means for achieving business objectives. While most of the effort in the field of information systems engineering is focused towards complex problems and larger enterprises, the specific needs and problems of smaller enterprises are often forgotten.

The problems we specifically look at in SMEs are the need for a better documentation, understanding, and analysis of the processes (know-how), why things are done (know-why), what concepts are used (know-what) and who is responsible (know-who), taking into account the specific characteristics of SMEs that may impose constraints on potential solutions.

In this paper, the goal oriented requirements engineering technique KAOS is proposed as a solution to document SMEs' know-how in process models, know-why in goal models, know-what in object models and know-who in responsibility models, and to make sure these models are aligned to achieve internal consistency and traceability. The paper describes the relevant characteristics of SMEs and their CEOs and evaluates KAOS in terms of how well it addresses these specific characteristics and needs of SMEs. An example illustrates how KAOS can be used in an SME and a case study gathers questions of an SME's CEO regarding KAOS and its tool Objectiver while building his own models.

The example given for an existing SME delivered some insights. First, the KAOS goal model enables SMEs to document their know-why by asking why-questions (justification) and how-questions (refinement). Alternatives can be expressed by OR-refinements, conflicts by conflicting goals, and obstacles can be analysed and resolved to make the goal model more robust. Second, SMEs' know-how can be expressed by means of a KAOS operation model, which has the extra advantage that the rationale behind the processes can be expressed by linking the operation model with the goal model via operationalization links. In this way, bi-directional traceability between problem and solution spaces is being assured. Third, an SME can make an internal or external agent responsible for goals and for performing operations. Fourth, the KAOS object model provides a common glossary.

This example showed that KAOS, as it was originally developed to be used in software system development projects, has the ability to document and analyse an SME's business architecture.

In the case study, the SME's CEO was very satisfied with the way in which KAOS and Objectiver enabled him to analyse his enterprise and to document both know-how and know-why. However, when building his model, he had some questions that provided us with material to work on in further research.

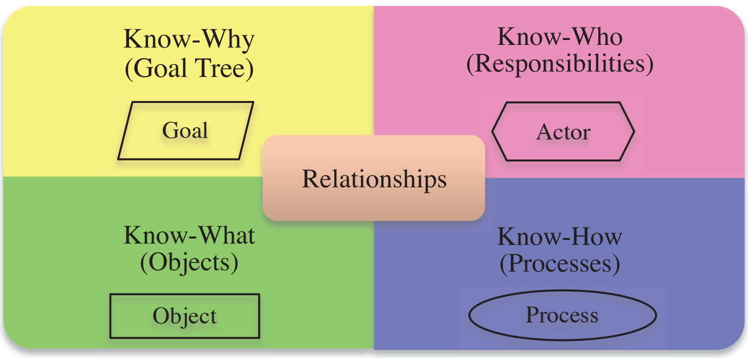


Figure 1: Structure of the four submodels of CHOOSE

After the publication of this paper, the results have lead to the development of the CHOOSE approach (Figure 1), which will be published in the special issue of *Informatie* on architecture in November, 2011[1].

---

[1] To be published as Bernaert, M. (2011): De Zoektocht naar Know-How, Know-Why, Know-What en Know-Who: Architectuur voor Kleinere Bedrijven in Vier Dimensies. In: Informatie, Sdu Uitgevers bv (2011), http://www.informatie.nl/

Towards an Investigation of the Conceptual Landscape of Enterprise Architecture*

Dirk van der Linden[1], Stijn Hoppenbrouwers[2,1], Alina Lartseva[3], and Erik Proper[1]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
{dirk.vanderlinden,erik.proper}@tudor.lu
[2] Radboud University Nijmegen, the Netherlands
stijnh@cs.ru.nl
[3] Donders Institute for Brain, Cognition and Behavior, Nijmegen, the Netherlands
a.lartseva@fcdonders.ru.nl

Abstract. In this paper we discuss the preliminary phase of our investigation into the conceptual landscape of Enterprise Architecture (EA). EA involves the creation of a holistic enterprise model which requires the integration of models describing many aspects and concerns. These models are often created by different communities in different modeling notations (e.g. i*, BPMN, e3Value, UML). Our goal is to aid in the validity of such integrated models by ensuring the semantics originating from the individual notations are well understood and not just superficially handled. Having a more fine-grained understanding of the semantics of these individual notations and how they are used by their respective modelers helps ensure that the enterprise model is a valid reflection of all the separate aspects. In order to do so it is necessary to explicate the semantic differences between the constructs of these notations as well as between communities using the same constructs differently. To accomplish this we selected a number of modeling notations and related methods that cover a wide area of use in academia and industry. We distilled the semantics of the constructs from their official (or most widely accepted) standard or specification. Following this we classified each construct as detailed as possible, after which we iteratively clustered them into a common category. When (superficially) similar constructs diverged in their categorization we denoted the relevant discriminating semantic factor. This was repeated until all constructs were categorized in a minimum amount of categories that were still domain specific (i.e. stopping short of categorizing constructs purely as ENTITY or RELATIONSHIP). The results from our analysis show that there is a common high-level categorization of concepts shared between the different notations and communities, although certain notations lack constructs for some categories (e.g. most GOAL-oriented notations not having constructs to describe results). The results also show a small number of discriminating factors (e.g. necessity, intentionality, materiality) that correlate strongly with

* Originally published as: van der Linden, D.J.T., Hoppenbrouwers, S.J.B.A., Lartseva, A., Proper, H.A.: Towards an Investigation of the Conceptual Landscape of Enterprise Architecture. In: T. Halpin et al. (Eds.): BPMDS 2011 and EMMSAD 2011, *LNBIP* 81, pp. 526–535. Springer, Heidelberg, 2011

the focus of a notation or community. These can be used to characterize their respective understanding of some constructs. While the categories of individual notations tend to have a different central focus (e.g. focusing on material or immaterial RESOURCES), they are capable of being abstracted to the same semantic component. Furthermore, many of the notations which are focused on a specific aspect tend to have a greater number of constructs for aspect-specific important concepts that allow them to express a more fine-grained semantics for those concepts than their widely-scoped counterparts. These results support the idea that investigating the detailed personal semantics of different notations used in EA is necessary to ensure the semantic consistency and validity of enterprise models.

Keywords: enterprise modeling, conceptual landscape, ontology, category, prototype

Issues and Challenges in Dynamic Systems Design and Engineering – A Value-Oriented Approach

João Pombinho[1], José Tribolet[1,2],

[1]CODE - Center for Organizational Design & Engineering, INOV, Rua Alves Redol 9, Lisbon, Portugal
[2]Department of Information Systems and Computer Science, Instituto Superior Técnico Technical University of Lisbon, Portugal
jpombinho@acm.org, jose.tribolet@inesc.pt

Abstract. Modeling organizations as complex systems in permanent evolution, as an answer to change dynamics, is an increasing challenge. Particularly, there is a lack of an integrated perspective that is generally and recursively applicable to organization chains, organizations and sub-organizations of several types and sizes. Our research aims to answer how to incorporate purpose into system development activities, in a way that promotes value-orientation and innovation. Three main conceptual challenges were identified: 1) the lack of capacity to view a system, and the services it provides, integrated in different value chains; 2) the separation of the instance of a system from the purpose behind its design; and 3) the conceptual unidirectionality of the system development process. In this paper, we present the proposal of rationalizing system design and engineering decisions with value-orientation, materialized in a set of principles and a four-layer framework: System, Service, Market (Value) and Problem Solving (Purpose).

Keywords: System Design and Engineering; Demand Management; Purpose; Market; Value Chains; Service Oriented Architectures.

1 Introduction and Motivation

The main premise of Organizational Engineering is that organizations are systems and, therefore, can be object of engineering activities. ICT-based organizations are especially promising candidates for this kind of approach since their processes are mostly immaterial, ranging from a few activities to the whole chain and even the final product or service. Events are generated and handled in ways that facilitate their capturing in comparison to other systems without explicit state representation. But regardless of the main type of agents that support the organization activities, the focus should turn from *doing things right* to *doing the right things*, as it is inglorious to have outstanding performance at something that should not be done at all. This is especially critical in ICT-based organizations, where the high level of automation allows for transactions to be executed massively. This fact amplifies any flaws in the creation process of such systems, which end up embodying requirements that are implemented without being formally aligned into an overall model. Even if the initial implementation serves the purposes it was created for, the evaluation of impacts, conception and implementation of subsequent changes is difficult to perform in a rational manner due to modelling shortcomings.

In order to tackle these classical issues from an innovative perspective, we begin by asserting that every kind of organization, regardless of their composition and objectives (private or public, political, business, education, healthcare, non-profit, etc.) brings about some form of value, directly or indirectly, so this is a unifying concept. Also, a given system is *one possible solution (out of many) to a problem*; a *means*, not an *end*. The market does not request an organization; instead, it values the services that it provides and that contributes to a solution for a given problem. Therefore, it is the organization that should reconfigure itself as a system to have the capacity of providing the services requested by the market, not the other way around.

Formal organizations are generally created as providers of a *repeatable* and *stable* solution to a demand, meaning there is reasonable belief that its elements will be continuously available. The rationale behind this quest for stability is, essentially, the lack of agility in procuring resources on-demand, compromising between evaluating every possible solution to each business activity and the time and effort consumed in doing so. However, with the current change pace, stability is a luxury unavailable to most organizations as the demand set itself changes. Therefore, a framework must explicitly include the concept of market, with demand/offer dynamics.

In addition, being market-aware means *recognizing the user's freedom of choice* – in the end, in every chain there will always be an end-user! Even in operational dynamics, it is frequent that people use alternative, unofficial means of performing actions; not recognizing it as a choice, in a formal or informal market, is missing the opportunity to improve organizational design and engineering.

This paper reflects ongoing research and is structured as follows: Section 2 presents problem analysis with a motivation example from a Library DEMO model, which is the base for identifying current challenges. These are grouped in five problem areas, with the corresponding research questions and a brief and localized related work review. In Section 3, we present a set of principles currently applied in a real-world setting to tackle the identified issues, along with a Framework overview. The paper closes with conclusions and contribution summary in Section 4.

2 Problem Analysis

2.1 Base Theory: Systemics, DEMO and the GSDP

This paper addresses system development from a problem-solving perspective driven by value. The *system* definition we will use, from [1], defines the following properties for a system: *composition* – a set of elements of some category; *environment* – a set of elements of the same category, disjoint from the composition; *production* – things produced by elements in the composition and delivered to the environment; and *structure* – a set of influence bonds among the elements in the composition, and between them and the elements in the environment.

Design and Engineering Methodology for Organizations (DEMO) [1] is a cross-disciplinary theory for describing and explaining the structure and action of organizations. It defines an organization as a discrete dynamic system consisting of social actors, who enter to and are responsible for commitments with each other in a coordinated manner. Enterprise ontology is a model of an organization in which these commitments serve as models for business transactions. DEMO was chosen because it models the essence of transactions between responsible actors and abstracts away implementation issues. However, it is currently not widespread in terms of awareness by the community. Included in its theory set is the *Generic System Development Process* (GSDP), shown in Fig. 1, which begins with the need by a system, the *using system* (US), of a supporting system, called the *object system* (OS).

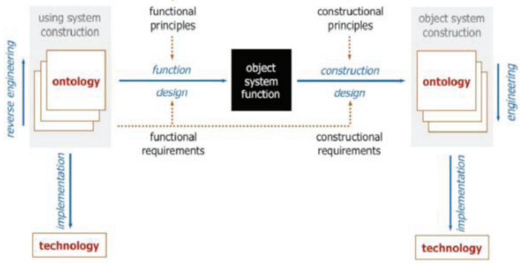


Fig. 1. Generic System Development Process [1].

From the white-box (WB) model of the US, one determines the functional requirements for the OS (function design), formulated in terms of the construction and operation of the US. Next, specifications for the construction and operation of the OS are devised, in terms of a WB model (construction design). The US may also provide constructional (non-functional) requirements. Choices are then made with each transition from the top-level white-box model towards the implementation model.

The GSDP has articulate and clear primitive concepts that reflect the essence of system development. We chose to use it as a reference, since we believe the critical analysis is extensible to other system development processes.

To close this brief presentation of the base theory set, it is important to differentiate two aspects of a system: *Teleological*, concerning its function and behaviour, a black-box; and *Ontological*, about its construction and operation, a white-box [1]. The main question is: *How to integrate the teleological and ontological conceptions of a system so that proactive innovation and value-orientation is promoted?*

2.2 Current Challenges Identification and Analysis

In order to clarify the problem space, constituted by a large set of core concepts from different concern areas, a practical scenario based on the classical DEMO Library case [1] will be used for instantiation. In this example, the elements of the system dealing with the membership (solid black line-bounded area in Fig. 2) are not justifiable as bringing direct value to the customer, who only wants to get hold of a book. However, as it can be seen in Fig. 2, this is all but clear in the ontological (construction) model:

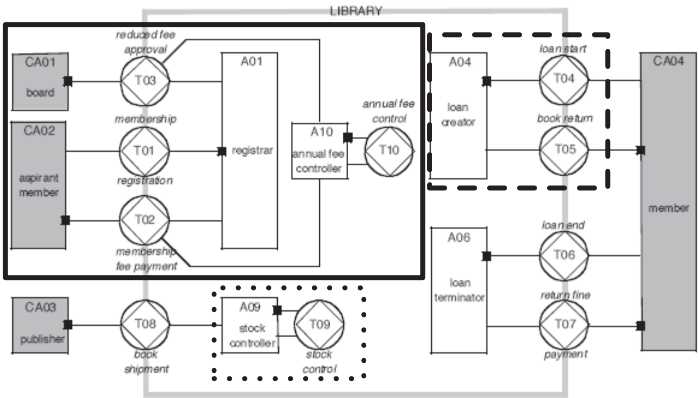


Fig. 2. Library example – Construction analysis.

Regarding the core business of *providing reading content*: 1) the *core service* is concealed in the area marked by a dashed line, obscured inside a *loan transaction*; 2) inside the solid black line, a *sacrifice* of the customer in obtaining the service and its support (sub)system; finally, the area bounded by points encloses a support process that may need revision, for instance, in a change scenario of going digital.

About the Membership Management subsystem, one must ask if there is really a customer who *wants* a membership or was this subsystem included in the Library as the manifestation of a strategy to get a fixed amount of income to face, for instance, stocking management? Is this still a problem if the organization does not pay for the books and space? Is it done for profit or simply as a response to the cost of keeping a large library? Is it part of the Library concept, i.e., every library also offers it by definition? Under what conditions should this **decision** be **reviewed**?

There are a number of approaches of different nature to parts of these problems, including system development by Dietz and Hoogervorst [1], Service Design by Bell [2], Enterprise Architecture by Lankhorst [3], Goal-orientation [4, 5] and Value Management by Gordijn [6] [7], to name a few; however, none of the questions can be answered directly by these or any other framework that we are aware of.

By analyzing the current State of the Art, the following five problem areas were isolated, with their respective Research Questions:

2.2.1 Value Definition

Value is, by nature, dependent on the stakeholder and, thus, relative. The problems in adequately naming and scoping of a service, know in the Service Design community, are a symptom of this [2]. Regarding the Library's purpose, what is the core transaction for providing value? For instance, should the transaction be named "Loan book" or "Provide (limited-time) access to (reading) content"? Is the "Membership registration" service is interesting *per se*, or is it only in the way of getting a book, that is specific to this particular construction of a library? This is why current goal-oriented modeling [4, 5] is not enough: it lacks an independent value structure to refer to. It must be understood that this structure is not subordinate to the service-providing systems, but the other way around! e3Value [7] provides essential value mapping perspective but lacks a holistic and formal framework for enterprise modelling.

RQ 1: *How to 1) represent* ***value*** *as a manifestation of* ***purpose****, in a structured yet relative way and 2)* ***trace*** *it through system development deliverables?*

2.2.2 Value Production Semantics and Business Model Definition

Systems design and engineering activities are guided by principles and requirements, normally based on informal specifications such as textual descriptions of use cases. A system's production is the best alignment beacon as it is the effective contribution to its environment. Current approaches do not model the system's production in a way that can be engineered. In our example, the same construction would serve both a Book and a Music Library; is the loan mechanism an interesting way to provide both types of content? Also, what is the threshold where an organization ceases to be of a certain type and what are the more general and specific organization types? Business Model Canvas [8] is an interesting and pragmatic approach that shares this concern area but lacks the formality that allows effectively entering the system engineering phase. For instance, a Library without a Membership subsystem is still a Library; but is it still a Library without a Catalogue?

RQ 2: *How to represent the* ***semantics of a system's production*** *in a relevant way and how does it contribute to the* ***essential definition*** *of a system?*

2.2.3 System/Sub-system (De)Construction Modeling Support

The construction of a system resulting from the development process is a compiled structure that obscures the system/subsystem relations and their motivation. It is very hard to separate a given subsystem from its owner system, especially if it was modeled from a flat description of the operation of the organization, instead of a sequential bootstrap or an incremental design step.

Assuming the stability of a value chain is generally unsafe because of change dynamics, which justify the need for a structure where to represent multiple scenarios in order to provide a flexibility point instead of a frozen solution path. How does a Library compare to a Bookstore or a Publisher, from the customer's perspective?

RQ 3: *What concepts are needed to define* ***system and subsystem relations*** *so that they can be applied* ***recursively****? How to represent* ***multiple scenarios*** *regarding different solutions to a given problem in a flexible way?*

2.2.4 Lack of System Intervention Rationale Modeling

It is quite common that questions about system intervention rationale are very hard to answer, especially some time after it has happened. For instance, regarding the introduction of the Membership subsystem: 1) When was the decision taken? 2) What was its purpose? Was it for mitigating the risk of non-return? 3) What were the design principles, constructional principles, assumptions and constraints applied? Are they still valid? For any kind of content the library may want to provide, e.g., e-books?

DEMO has been extended [9] to incorporate change dynamics but, at this time, still does not model the formal rationale of each change. This is particularly relevant in creating new, innovative, components of the organization, both in bootstrap and in on-going phases. The GSDP also does not prescribe what to do with the objects supporting the rationale of the decisions made during the process. The implementation steps consist in introducing restrictions on the construction, for instance: 1) assumptions, such as assuming the customer is necessarily a reader; 2) constraints, such as available technology to offer books, e.g., physical or digital.

RQ 4: *How to define the **rationale of a decision** in terms of the application of design principles, constructional principles, assumptions and constraints in a **structured** way that is relevant and explicitly include it in the system model?*

2.2.5 Conceptual Unidirectionality of the System Development Process

The unidirectionality of the system development process induces an upper limitation of the solution's value, indexed to the original functional request scope. Extra value that could be derived in bottom-up fashion, either available at the original design time or in future interventions, is not addressed. According to the GSDP, *Determining Requirements* is defined as 'The design phase that starts from the ontological model of the using system, and ends with the functional model of the object system' [1]. This approach requires full knowledge about the US, which is a serious limitation. Even if it were trivial, the solution would be irrecoverably restricted to satisfying the demand of a specific US, its value is limited from the outside instead of being allowed to expand creatively inside out. This is why the Agile [10] paradigm does not fully solve this issue, regardless of the length or frequency of the development cycles.

Again using the Library case, if e-books begin to be provided by the Library system, what are the possible USs for that new OS? For instance, a Printing on Demand (PoD) service requires no stock control of physical books.

RQ 5: *What is the process of supporting **innovation** regarding the essential definition of a system, and which concepts result from the introduction of **bidirectionality** in the **GDSP**?*

3 Towards a Solution: Principles and Framework Overview

3.1 Principles of a Different Way of Thinking

In this research, we are proposing a set of principles that were derived from practical application at a real-world Demand Management scenario:

- Recognize the system being developed as **one of many possible solutions** for a problem and, therefore, as a **means**, not an end;
- Conceptually integrate the **Teleological** and **Ontological** perspectives of a system by introducing the problem/solution paradigm and value concepts into system modeling activities;
- Improve **problem definition and elicitation** by using the concepts of system value, subsystem value generation and positioning the system in a demand/offer relation between consecutive nodes in a **value chain**;
- **Improve the clarity of system models**, by embedding value-semantics in the development process and **tracing** it to the relevant system elements as a structured means of expressing **purpose**;
- Look beyond the boundaries of formal organizations, into value nets, as a provider may serve multiple customers (n-1) with different problems and expectations, assisted by multiple suppliers (n+1) to increase design abstraction so that system value is increased as a result of greater market;
- Improve **change evaluation and decision rationale** by applying design principles, constructional principles, assumptions and constraints in a relevant, structured way that is explicitly included in the resulting model;
- Support Innovation by using these intermediate constructs from the development process to conceptually reverse the development process in a rational way during a reengineering effort – Reverse Discovery.

This set of principles reflects the current thinking and results, and in the course of research will be further refined and validated. It is important to note that it does not imply a specific way of working and is independent of tool support - even though it can be greatly aided by it, especially according to portfolio size and change rate.

3.2 Framework Overview

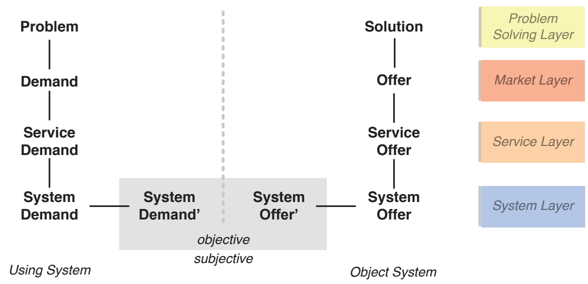


Fig. 3. Framework Overview

Our high-level solution proposal includes a four-layer framework: System, Service, Market (Value) and Problem Solving (Purpose); their relative positioning is represented in Fig. 3. The most differentiating concepts for each of these layers are presented next, in bottom-up fashion: from *system* towards *purpose*.

3.2.1 System Layer

For the purposes of this paper, we will focus on the *recursivity* property of our system definition. We argue that any given complex system can be decomposed into more granular systems chained together; the rationale for forming each link is the same that should exist between the components of a system for, in the end, the same concepts will recursively apply. We base the last statement in the following assertion:

If a single element is part of a system's *composition*, then it is connected by means of the system's *structure* to other elements; therefore, this connection must represent (but does not necessarily specify) the element's contribution to the *production*.

A single element of a system is also a system (a sub-system of the original system), with a *composition* constituted by a single element, an *environment* formed by the other elements in the original system, a *structure* linking the element to the environment and a *production* as the fact pertaining the *contribution* it makes to the production of the original system – which is its *purpose*, regarding that chain.

3.2.2 Service and Market Layers

These two layers are responsible for mediating the relation between a customer and the systems that participate in solving his problems. The service layer abstracts functionality from a given conceptual system in terms of inputs and outcomes while framing it in transactional semantics, with exchange of contract and operation conditions. In turn, the market layer uses value as a driver to procure and assemble service sets complying with the solution to a given problem.

Returning to the example, the Obtain Book Service abstracts away any implementation choices or provisioning mechanisms. Hence, it brings the Library's production to an essential level that puts them all in the same level, which is the first step in allowing comparison to other alternatives of bringing about such item. Some examples are online ordering, loaning at a library or borrowing from a friend. Each of these variations introduces an offer at the solution market level with specific pricing and dependencies, which end up providing different end-user experiences.

While organizational-centric modelling may seem more natural because of its formal boundaries, the service structure is arguably more important since it is, by definition, focused in performance and value creation. This happens even at an intra-organizational level, as each sub-system is a means for providing value through services. Due to space constraints we will not define the structure of individual services in this paper, but we refer the reader to [11], where a framework is presented for service specification based on enterprise ontology.

3.2.3 Problem Solving Layer

Essentially, this layer is responsible for defining the problem statement and matching it to solutions available in the market. These solutions are sets of services that are contextualized and presented as value exchange propositions. In order to perform meaningful modelling and reasoning it is essential to establish the *purpose* as it is the base for designing and engineering the solution providing system. Purpose is: ‘(...) an object or end to be attained; what one intends to do or bring about’, according to the Merriam-Webster dictionary. A system’s purpose is hard to formalize as stakeholders frequently formulate a high-level solution instead of the real problem, or present it in ways that induce specific solutions, such as in the classic example by Henry Ford: “*If I had asked people what they wanted, they would have said faster horses*”.

Language and problem formulation is also critical as it drives the definition of the elements of the solution set [12]. An interesting model for its formal explicitation is presented in [13]. It consists of Need, Want and Demand structured in an hierarchy consisting in a transition from a *need* - a problem statement – to a high-level solution, defined as set of services that together provide a solution for that need - a *want* - and then to the formulation of a *want* in terms of value exchange proposition - a *demand*.

There are two other significant obstacles to problem solving, from the set identified by Mayer [14], that we are interested in tackling in this research:

Functional Fixedness: the tendency to view problems only in their customary manner, preventing vision over different options that might be available to find a solution. This is directly related to the upstream ramifications in a value chain.

Assumptions: when dealing with a problem, assumptions about the constraints and obstacles are often made, preventing certain solutions.

Both are conceptually addressed by using Reverse Discovery and recursion in the application of the problem solving to each engineering step.

4. Conclusion

This paper presents the current research results as an overview of a complex and largely subjective problem space. The presented structure and methodology is deliberately generally applicable to any human-engineered system, not only organizations, making it a very ambitious modeling effort in terms of abstraction.

During literature review, we were unable to find any framework structured in a way that solves the identified problems. We are confident that they are extremely relevant since they can be reiterated at any system/sub-system relation, either at pure business level, business/ICT interface or inside complex ICT systems. The abstraction and flexibility enabled by the recursive application are especially relevant in ICT-intensive environments, as the access to components usable as pieces of a solution chain is increased and maturing technological advances, such as the Cloud, make real-time service market start to look plausible in a relatively short timeframe.

Our contribution is composed by: 1) the identification of a relevant problem space in current approaches (both in academia and industry), particularly the lack of a sound structure to model purpose and serve as an ongoing referential, instead of addressing it solely at the early stages of individual system development cycles and losing track of it afterwards; and 2) the definition of a *conceptual high-level framework* that addresses, by design, the main issues identified in section 2 of this paper. It integrates the core concepts and their relative positioning in a layered manner, differentiating the concepts that characterize a problem/solution pair end-to-end, from need to implementation. The most important conceptual contributions are:

1 *Integrating the Teleological and Ontological perspectives* of system development by framing it in a *problem-solving context* and introducing the concept of *Market*;
2 Defining the *rationale of choices in terms of availability of solutions in a market*. This is accomplished by recursively defining purpose of a system as its contribution to that specific chain;
3 The *Reverse Discovery* concept as a different view over the GSDP, allowing structural accommodation of innovation dynamics.

Combining with Design Science Research, the methodology applied includes Action Research and has been adapted to a professional context in IS Demand Management, interfacing Business and IT at a leading Telco operator. Activities include analyzing motivation, impacts, cost vs. benefit, consolidation and planning of initiatives. Additionally, we have modelled part of the framework in formal ontology and build a Protégé-based prototype for supporting a preliminary case study, which has been used for instantiation of real world scenarios and was instrumental in eliciting hidden value assumptions obscured by upfront, unguided, service design.

References

1. Dietz, J.L.G., *Enterprise Ontology: Theory and Methodology*. 2006: Springer.
2. Bell, M., *Service-Oriented Modeling: Service Analysis, Design and Architecture*. 2008, New Jersey: John Wiley & Sons.
3. Lankhorst, M., *ArchiMate Language Primer*. 2004.
4. Lamsweerde, A.V. *Goal-Oriented Requirements Engineering: A Guided Tour*. in *International Symposium on Requirements Engineering*. 2001. Toronto.
5. Regev, G. and A. Wegmann, *Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering*, in *13th IEEE International Requirements Engineering Conference (RE'05)*. 2005, IEEE: Paris, France.
6. Gordijn, J., E. Yu, and B.v.d. Raadt, *e-Service Design Using i* and e3value Modeling*. IEEE Software, 2006. **23**: p. 26-33.
7. Gordijn, J., *Value-based requirements Engineering: Exploring innovatie e-commerce ideas*. 2002, Vrije Universiteit Amsterdam: Amsterdam.
8. Osterwalder, A. and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. 2009: Self-Published.
9. Aveiro, D., *G.O.D. (Generation, Operationalization & Discontinuation) and Control (sub)organizations: a DEMO-based approach for continuous real-time management of organizational change caused by exceptions*. 2010, UTL: Lisboa.
10. Beck, K., et al. *Manifesto for Agile Software Development*. 2001 [cited 2010-11-28]; Available from: http://agilemanifesto.org/.
11. Albani, A., et al. *Enterprise Ontology Based Service Definition*. in *4th International Workshop on Value Modeling and Business Ontologies (VMBO)*. 2009.
12. Schooler, J.W., S. Ohlsson, and K. Brooks, *Thoughts beyond words: When language overshadows insight*. Journal of Experimental Psychology, 1993(122): p. 166-183.
13. Kinderen, S., J. Gordijn, and H. Akkermans. *Reasoning about customer needs in multi-supplier ICT service bundles using decision models*. in *11th International Conference on Enterprise Information Systems*. 2009. Milan.
14. Mayer, R.E., *Thinking, problem solving, cognition*. 2nd ed. 1992, New York: W. H. Freeman and Company.

Session 2: Service Science

Agile Process for Integrated Service Delivery

Marjana Shammi, Sietse Overbeek, Robert Verburg, Marijn Janssen, and Yao-Hua Tan

Faculty of Technology, Policy and Management, Delft University of Technology,
Jaffalaan 5, 2628 BX Delft, The Netherlands, EU
mshammi@gmail.com,
{S.J.Overbeek,R.M.Verburg,M.F.W.H.A.Janssen,Y.Tan}@tudelft.nl

Abstract. Companies want to become more customer-centric and embrace Integrated Service Delivery (ISD) to provide a single shop. The integration of services of different organizations results in the creation of dependencies among services which are in different stages of the life-cycle. Only with effective collaboration between the parties and coordination of development activities, ISD can be managed efficiently. With the adoption of Agile methodologies, performance can be gained reducing the complexities of software development and focus on collaboration and coordination aspects. Therefore, this research proposes a model on how to manage the service lifecycle of ISD in a top-down view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile Scrum approach. The method is different from existing ones as it uses agile principles applied to life-cycle management and incorporating iterative development from the commencement of requirement analysis until the completion of the development of services.

Keywords. Integrated Service Delivery, Agile, Scrum, service, development process

1 Introduction

Companies are becoming more and more customer-centric: understanding and anticipating the needs of customers, designing what customers want, and then aggregating and managing the components and suppliers to rollout products and services quickly and cost-effectively to meet ever-changing customer needs. With the opportunity of Integrated Service Delivery (ISD), companies can support clients in an integrated environment possibly reducing cost and time. ISD can be defined as *'a bundle of services provided by a single service provider or multiple service providers collaborating with each other through a single interface accessible to clients'* [1-3]. Providing these services, service providers face a number of challenges related to organizational integration, resistance towards change and managing the dependencies among services. With effective collaboration and coordination of activities, ISD can arguably be managed efficiently. To support the development of ISD and the updating of services, companies look for agility in their development process of ISD. Research has shown that adoption of Agile methodologies has reduced complexities in software development and there is an increasing focus on collaboration and coordination to achieve performance gain [4]. Agile development processes are characterized by incremental and iterative software development by teams closely collaborating together[5]. To the best of our knowledge, there has not been research on how to manage the service lifecycle of ISD in a top-down view and focus on the collaboration of parties involved in the process and coordination of activities, by working in an Agile development approach. Researching this aspect can provide an insight on the iterative perspective of the process and help companies to incorporate and benefit the best practices out of it, to effectively collaborate and coordinate. This research aims at understanding how the *Agile management principles* can further be blended with the *service development principles* and be incorporated throughout the lifecycle to focus on the collaboration and coordination in ISD management. Thus, this research proposes such a process - *Agile Process for Integrated Service Delivery (APISD).* Compared to traditional software development models such as Waterfall, Spiral and incremental development models, the APISD model introduces the iterative development from an earlier stage. This is because it is equally important to invest time and effort in proper requirement analysis and designing just as in development. Early iterative development allows adapting the changes flexibly compared to adapting them at a later stage. Moreover, incorporating the iteration allows the respective teams to work based on priority and produce usable artifacts in short periods of time. Furthermore, this model is different and extends from existing Agile development methodologies, because it envisions a wider focus at the entire lifecycle instead of focusing only on the development phase; methods such as Extreme Programming, Test Driven Development, Feature Driven Development and Scrum itself does [6].

To conduct this research, a design science research methodology has been employed as the research approach and case study research as a research strategy. The design science approach [7] was chosen since it addresses important problems that can be solved in an effective way with the help of an innovative artifact provided in this research. Case studies were investigated by reading reports and conducting three interviews with three organizations. Six steps have been followed, which are: Problem Identification and motivation, Definition of objectives, Design and development, Demonstration, Evaluation, and Communication [7]. The structure of this paper is as follows. Section 2 consists of a literature background on the concepts followed, section 3 consists of the derivation and description of the developed conceptual model and finally section 4 provides conclusions for this research and future work.

2 Literature Background

The following section briefly discusses the theoretical groundwork that was covered on the two concepts of ISD and Agile methodologies.

2.1 Integrated Service Delivery

When defining a service, there are many definitions that are based on technology or originate from the marketing literature. Some definitions are of electronic services, some thought of as web services, others are viewed as abstractions of business processes and some are considered to be an aggregation of other services [8]. Considering the various aspects surrounding the meanings of 'service', for this research, we contemplate the definition of service. [9], which is "*a series of* *interactions between the service provider and clients that result in an observable output*".

As far as multiple service providers are concerned to provide the integrated services, clients perceive a bundle of services provided by various service providers as a whole and do not have to deal with each single provider. The essence of this problem of ISD is that these services need to be integrated; however, they are often heterogeneous and not designed for this purpose. Therefore, understanding the challenges faced in ISD, service characteristics and the process of developing these services in a structured manner is important. To develop the integrated services, service providers face a number of challenges which are related to organizational integration, embracing change and customer satisfaction. In the case of *organizational integration,* challenges include addition of staff working under different work processes, standards or different collective agreements in case of multiple organizations [2]. Therefore, there is a need of a common language and vision. For effective collaboration, it is important for parties to agree and to set common goals, establish common assumptions and build trust in the beginning of the development lifecycle. Effective communication, a shared understanding of roles and responsibilities, and a collaborative method of resolving issues are considered to be key factors in a successful partnership [2]. When concerning *embracing change*, the reality in ISD is about change and that change requires a certain level of risk. To deal with the risks and adapt to changes, working in this type of environment requires extensive communication and coordination of activities to manage those changes accordingly. By embracing change and integration, companies can innovate and advance rapidly [2]. As for *customer satisfaction* - ISD must be driven by a common desire to increase customer service. ISD partners should seek to satisfy stakeholders by determining how to meet their needs and then actually meeting them [2]. To be a customer-centered organization, the organization should consult the customers and other key stakeholders on an ongoing basis. As the nature of ISD is customer service oriented, not addressing to customer needs will cause organizations to lose the competitive advantage [2] and decline their growth in the market.

In this research, we have studied the service lifecycle of services suggested by several authors. The purpose of these service lifecycle models are either to introduce a new approach to deal with the lifecycle management, which consists of new roles and new development tasks as opposed to the ones of traditional software engineering[10], [11], or to deal with the heterogeneity challenge in platform specific or independent functionalities[12]. There are also several models of service lifecycles used by various companies and according to Gu and Lago [10], that covers the organizational process flow of a service lifecycle with a relation between stakeholders and service lifecycle stages. From the investigation, these models have allowed us to understand and follow a theoretical perspective of the service lifecycle provided by Gu and Lago [10] and the phases suggested by Papazoglou and Heuvel [11]. The lifecycle consists of three phases, *design, runtime* and *change* [10]; where design refers to the lifecycle of a service before it is available for use; runtime refers to when services are put into production and the implementations start to work; change focuses on the life cycle of a service when adjustments have to be made when business requirements change. Within these phases, sub-phases mentioned by Papazoglou and Heuvel [11] exist: *planning, analysis and design (A&D), construction* *and testing, provisioning, deployment, execution and monitoring*. The roles involved throughout the service development are service provider, service broker and service consumer. These roles along with the phases were explored.

2.2 Agile-Scrum Methodology

Agile software development is a group of software development methodologies based on iterative and incremental development, which was termed and introduced by 'The Agile Manifesto' [13]. Some important characteristics of this manifesto are: (a) client satisfaction by rapid delivery of useful software; (b) welcome changing requirements; (c) working software is delivered frequently (weeks rather than months); (d) sustainable development, able to maintain a constant pace;(e) close, daily cooperation between business people and developers;(f) continuous attention to technical excellence and good design; and (e) regular adaptation to changing circumstances. One of the methodologies followed in the Agile software development is Scrum. The Scrum approach basically focuses on managing the system development process. It does not define specific software development techniques for implementation but rather concentrates on how team members should function to produce a system adaptively in a constantly changing environment. The characteristics of Scrum have been provided by Schwaber [14]. These are: flexible deliverables, flexible schedules, small teams, frequent reviews, inter and intra-collaboration, object oriented development. According to Schwaber and Beedle [15], the lifecycle consists of three phases: *Pre-game, Development and Postgame*. The roles involved in this lifecycle are: scrum master, product owner, scrum team, client, management and user; who were described in details.

3 Defining the Model

After understanding the characteristics and lifecycle of ISD and Scrum, we developed a conceptual model. The following section briefly elaborates on the model itself, how we evaluated it and finally how the model can be used in practice.

3.1 Model Construction

In order to construct the model we looked into the commonalities of ISD and Scrum. We tried to determine the phases for APISD by amalgamating the phases of ISD and Scrum creating a mapping between them. Similarly, the roles required were defined, which were required for APISD, and were inspired from ISD and Scrum. With the necessary components derived the model was developed. As shown in Figure 1, the model is a lifecycle consisting of six phases derived from the amalgamation of ISD and Scrum given in section 2.1 and 2.2: *Planning, Service Modeling, Service Construction, Provisioning, Deployment and Execution and Service Management.* Activities within each phase were described on how the process will be performed and focused on how to overcome the challenges. As explained in the Introduction, this model includes practices from the Agile principles but is different from other Agile development methods because first, it does not focus only on the development of the services but instead on the whole lifecycle and second, unlike the other methods, this method incorporates iterative development starting from the requirement analysis. The early iterative development allows the requirement analysis and the designing to be considered a development process themselves in its nature, thus enhancing the clarification and the adaptation of changes at an early stage.

The *planning* phase consists of activities that allow businesses to analyze the business needs and market requests, and to determine the vision and objectives. With that knowledge, businesses are able to identify the type of services required and to be provided. The planning phase is carried out by the *service board*. The service board meets with the *client* and discusses various aspects of the services to be developed. The service board drafts a project document. They deliver this document to the *service analyst team* who will start with the analysis of the project. In the case of multiple service providers in serving the board, they also draft SLAs for their own governing responsibilities. This activity is crucial, because if the responsibilities and understanding between the parties are not addressed or agreed upon, several problems related to miscommunication, lack of ownership, and lack of coordination will arise throughout the lifecycle[11]. As a result, service providers will not be able to collaborate smoothly or gain trust, which is arguably required for sustainable development.

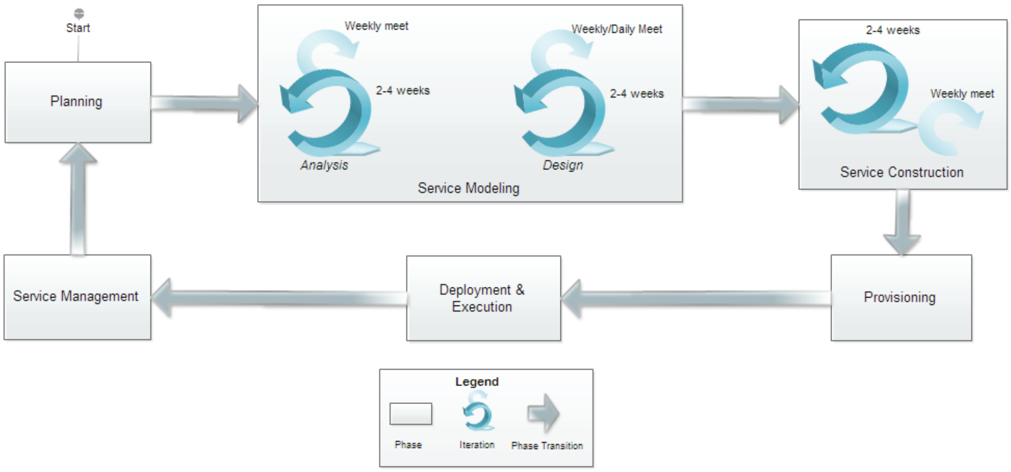


Figure. 1. APISD Lifecycle.

The main objective of the *service modeling* phase is primarily to describe the services identified in the planning phase consisting of two sub-phases, Analysis and Design. The team comprises of *service analysts* in Analysis, *development managers, and chief service developers* of the corresponding service providers associated in the project in Design. A *domain manager* exists which is appointed by the service modeling team who manages this phase to ensure that the different activities of analysis and design are aligned. The artifacts produced from this phase are: service backlog consisting of the services required and their requirements; feature backlog consisting of features derived from each service and their requirements and assigned to construction teams; and technical designs produced by the design team required for the implementation of the services. This whole process continues in separate iterations with the corresponding set of prioritized services. This phase is different from current practices because the iterative process begins at an earlier stage than the actual development. Moreover, the distinction between the requirement specification and technical designing from the actual development of the services in an iterative form, allows adapting to changes flexibly and focusing on prioritized work rather than implementing at one attempt.

The *service construction* phase consists of the actual development and ongoing testing that Agile methods suggest. First, the construction team(s) (*service developers, service testers, development manager(s)*) of either a single service provider or multiple service providers views the feature backlog set for the first iteration which lasts for 2-4 weeks. According to the assigned features, services are developed and tested. Considering the scenario with multiple service providers, solving integration issues will require each organization's developer to communicate with each other and solve. By communicating with each other, they are able to gather knowledge (which promotes collective growth) and coordinate effectively to solve the issues. Once all issues are solved, a demonstration of the integrated services is given to the client by the *development manager*. By involving the client at this stage, the service provider is able to acknowledge their needs and ensure those needs are met, as a result satisfying the client. The development manager handles any conflicts between the development and test teams. In order to coordinate effectively among the teams, the development managers of each provider meet weekly. In this meeting, they discuss any impediments, dependency related issues and further planning of iterations. The *release manager* meets with the service board to discuss releases and finalizes them with the development manager. Towards the end, the development managers also arrange a retrospective meeting of their own to discuss results, lessons learned and improvement points. This whole process continues in iterations with the corresponding set of prioritized features and is implemented accordingly.

As soon as the service package is ready to be deployed, the *provisioning* phase deals with settling on the various rules and regulations surrounding the service delivery which are defined by the service board together with the client in the form of Service Level Agreement (SLA). This phase is required before making the services available to the client, because for effective collaboration between the service provider and the client, there needs to be an understanding and agreement regarding the usage and charges of the services. After the completion of the provisioning phase, the services are ready to be *deployed and executed*. The *system administrator* performs the necessary activities and deploys the system in the production environment. Once in production and used by the *users*, in the *service management* phase, the integrated services can be monitored and ensured that all the services are running according to the rules and regulations set in the SLAs. Regarding the management of the technicalities, the system administrator is responsible for configuring, managing and troubleshooting the servers. This phase also consists of change management which is very important so that changes are managed well in order to ensure a smooth operation of business; these changes are logged in a *change request backlog* which is later prioritized by the release manager for further planning and implementation. Changes are logged in by the *customer service*. They also report incidents in the *incident backlog* which are also prioritized by the release manager.

3.2 Evaluation of the Model

Following the constructs of case study research given by Yin [16], the model was applied to organizations that develop software providing integrated services and are looking for a faster, flexible and structured way to produce their products. For this research three cases were explored. Two cases were performed on a single service provider scenario and the third case on a multiple service provider scenario. The multiple service provider and one of single provider have just commenced in following Scrum only in the development phase. The other single provider follows a Waterfall approach with moderations in the development phase. Semi-structured interviews were performed with a questionnaire where data was analyzed based on the answers provided in the questionnaire, interview discussion, audio recordings, website documentation and email correspondence.

In the interviews first the model was demonstrated to the organizations and evaluated with their current process. The comparison resulted into identifying differences between the two processes. From the analysis of the case study findings, additional factors were identified that have been used to enhance the model. As the *unit of analysis* is the implementation process to be investigated, which is a single unit of analysis, the case study takes towards a holistic view. Types of validity were looked at towards the case study findings. Construct validity was relevant because multiple sources of evidence were looked at providing a chain of evidence and further increased due to informants reviewing the draft case study report. Internal validity was irrelevant for this research because the nature of the case study was explorative instead of explanatory or causal. External validity was relevant because the cases were different and the model was replicated for all three which resulted in findings that can be generalized for other similar case studies. Finally, the reliability can be determined by following the case study procedure that was followed.

From the evaluation of the model, six key factors were identified and later appended in the model. (1) The service board was divided in two sub boards with a distinction in responsibilities serving a strategic and tactical nature, namely the *Executive Board* and *Service Board.* (2) The customer service was included in the Service Modeling phase to review the service backlog produced by the service analysts. (3) The system administrator was included in the Service Modeling phase to review the non-functional requirements defined in the Service Backlog and to provide input. (4) In APISD after the construction phase, a high level product demonstration is given to the customer service and system administrator. This way, these roles are acknowledged of how the services work and can better support the service management. (5) In case of rejection by the client after post-production, an activity was required included in APISD for analyzing the problems and producing possible solutions. (6) A workflow was required for the service analysts to also visit the users' work-floor and observe their interaction and engagement with the integrated services. In comparison to the existing methodologies followed in the cases, they have identified some advantages of the APISD model which are: incorporation of iterative development earlier in the phases from construction; creation of a separate phase regarding *provisioning;* division of the design phase from construction phase allowing focus on architectural decisions; detailed description of roles and responsibilities explicating the collaboration between the parties involved in the process; and coordination of ongoing activities between the service analysts for requirement specification and construction teams for service development.

3.3 Illustration of the Model

With the case study research findings appended to the model, the model was finalized. In addition, an illustration of the model was given on how this model can be practically implemented in an organization. Here, the implementation was based on two scenarios: for a single service provider and for multiple service providers collaborating together to deliver integrated services. Both scenarios were presented with a real life staging of the service provider and client and the type of services required. In each scenario, the activities within the APISD lifecycle phases were elaborated and the iterations were described on how team members can follow the iterations one after another maintaining synchronous information flow with other members. Examples were provided of how the service provider(s) collaborate(s) with the client and among themselves, what type of complications are faced and how they can deal with them using the constructs provided in the APISD model. Here, the concerns of the challenge of coordination were met by coordinating the following necessary activities in each of the scenarios: *decision points* were set- for example, once the service analysts produce the service backlog, only the service board decides and prioritizes the services. In the case of multiple-service providers, conflicts within the different teams are resolved by the development managers coming together to solve the dependencies or impediments; *change management processes-* to manage the changes, where a change backlog artifact is prioritized by the release manager, and implemented by the development team(s); an *issue management process* exists to manage the issues, where an incident backlog was introduced by which incidents reported by users are logged in, prioritized by the release manger and later implemented by the development team(s); *information sharing* was given importance, in order to have a consistent flow of information where teams are able to retrieve the requirements set in the service backlog and feature backlog and daily/weekly meetings were given within the iterations to share status and discuss impediments; finally, *performance review and monitoring* are portrayed- to monitor the progress, retrospective meetings were held by the company's development manager once the iterations are completed. These retrospective meetings comprises of lessons learned and identification of improvements to be implemented in future iterations. Moreover, other activities were detailed on what type of tools or artifacts the stakeholders can use while performing those activities. For example, in the scenario of multiple service providers, distributed teams can collaborate using virtual sharing tools such as TeamViewer or WebEx and designers can use collaborative diagramming using LucidChart. In order to fully understand the flow of activities among the different parties in the different phases, the implementation has also been demonstrated through sequence diagrams. Due to space limitations, these sequence diagrams are not part of this paper but can be looked at in [17]. These sequence diagrams provide a better understanding of how the various stakeholders interact with each other via the detailed activities per phase.

4 Conclusions and Future Work

The research objective was to provide an answer to the main research question on how Agile management and service development principles can be incorporated together for effective collaboration between parties and coordination of activities in Integrated Service Delivery. This was done by developing a conceptual model of Agile Process for Integrated Service Delivery (APISD).

With a literature analysis and from the author's experience, the model was developed to manage the heterogeneous services that are bundled to create integrated services. The model portrays *multiple service provider collaboration* where common goals and assumptions are established to build trust in the beginning of the lifecycle: the Planning phase. A detailed description of roles and responsibilities were specified that can provide a shared understanding, enriching the collaboration between the parties. The model also allows *change adaptation*, due to the nature of the APISD model having iterative development, demonstrating that the changes occurring during requirement specification and designing can be easily adapted in the subsequent iterations. Finally, the model promotes *client collaboration.* APISD enables service provider(s) to have close interaction with the client from the beginning so that they a continuously focus on the main desire of ISD, increasing customer service by being customer-centric.

From the evaluation of the model, key findings from the comparison and commended parts of the model were: importance of a separate phase regarding *provisioning;* importance of division of the design phase from the construction phase; detailed description of roles and responsibilities explicating the collaboration between the parties involved in the process; incorporation of iterative development earlier in the phases from construction; coordination of ongoing activities between the service analysts for requirement specification and construction teams for service development.

Finalizing the model has enabled us to answer the main research question. Moreover, both the industrial and scientific community can acquire an insight on the perspective of managing Integrated Service Delivery with Agile practices. The industrial community can develop an understanding of the iterative perspective of the process and incorporate the activities to collaborate with stakeholders and coordinate accordingly. Furthermore, they can try to adapt the process within their organization and incorporate customized practices to benefit their needs. The scientific community can critically analyze the intention of this research, the challenges that are dealt with, the method this research was conducted in, the model itself and perceive an understanding of the research findings. From the critical analysis, they can try to empirically test the model and identify improvements that can make the model stronger to benefit the organizations in the management of their technologies. Moreover, with that comprehension, they can try to investigate other methods of conducting this research for more efficiency and effectiveness.

The current research has several limitations and opportunities exist for solidifying the model. First the model can be actually implemented within an organization and empirically conclude that this process will result in efficient collaboration between the parties involved and coordination of activities in the APISD lifecycle. From the usage of this model in various organizations, the practicalities within the APISD process can be refined. Opportunities exist for delving in the acceptability of this model after implementation in organizations through extensive case study research and investigating furthermore into the accountability and governing mechanisms. Moreover, for further research, it can be investigated in the future of how a better case selection can be made specific to the type and number of service providers working together and the industry they are in. The evaluation of such specific selection criteria will be beneficial to empirically conclude on the effectiveness of the model focusing on collaboration and can be further generalized to broader sense of applicability. Finally, further research is necessary on the investigation of the research questions scientifically and it is recommended for researchers to publish new techniques and methods to implement this model and verify the effectiveness and efficiency of the implementation.

References

[1] L. M. Á. Sabucedo, L. E. A. Rifón, R. M. Pérez, and J. M. S. Gago, “Providing standard-oriented data models and interfaces to eGovernment services: A semantic-driven approach,” *Computer Standards & Interfaces*, vol. 31, no. 5, pp. 1014-1027, Sep. 2009.

[2] ICCS, “Integrated Service Delivery: A critical analysis.”Institute for Citizen-Centred Service, 2003.

[3] IAB, “Integrated Service Delivery Governments Using Technology to Serve the Citizen.”1999.

[4] J. Sutherland, G. Schoonheim, N. Kumar, V. Pandey, and S. Vishal, “Fully Distributed Scrum: Linear Scalability of Production between San Francisco and India,” in *2009 Agile Conference*, Chicago, USA, 2009, pp. 277-282.

[5] B. Boehm, “Get ready for agile methods, with care,” *Computer*, vol. 35, no. 1, pp. 64-69, undefined.

[6] P. Abrahamsson, O. Salo, and J. Ronakainen, *Agile software development methods : review and analysis*. Espoo [Finland]: VTT, 2002.

[7] K. Peffers, T. Tuunanen, M. Rothenberger, and S. Chatterjee, “A Design Science Research Methodology for Information Systems Research,” *Journal of Management Information Systems*, vol. 24, p. 45–77, Dec. 2007.

[8] J. O’Sullivan, D. Edmond, and A. Ter Hofstede, “What’s in a Service?,” *Distributed and Parallel Databases*, vol. 12, p. 117–133, Sep. 2002.

[9] M. Janssen, A. Joha, and A. Zuurmond, “Simulation and animation for adopting shared services: Evaluating and comparing alternative arrangements,” *Government Information Quarterly*, vol. 26, no. 1, pp. 15-24, 2009.

[10] Q. Gu and P. Lago, “A stakeholder-driven service life cycle model for SOA,” in *2nd international workshop on Service oriented software engineering: in conjunction with the 6th ESEC/FSE joint meeting*, New York, NY, USA, 2007, p. 1–7.

[11] M. P. Papazoglou and W.-J. Van Den Heuvel, “Service-oriented design and development methodology,” *International Journal of Web Engineering and Technology*, 2006. [Online]. Available: http://inderscience.metapress.com/app/home/contribution.asp?referrer=parent&backto=issue,7,7;journal,15,21;linkingpublicationresults,1:110898,1.

[12] L. W. F. Chaves, L. M. Sá de Souza, J. Müller, and J. Anke, “Service lifecycle management infrastructure for smart items,” in *Proceedings of the international workshop on Middleware for sensor networks*, New York, NY, USA, 2006, p. 25–30.

[13] Agile Manifesto, “Manifesto for Agile Software Development,” 2001. [Online]. Available: http://agilemanifesto.org/.

[14] K. Schwaber, “SCRUM Development Process,” *Proceeding of the 10th annual ACM Conference on Object Oriented Programming Systems, Languages and Applications (OOPSLA)*, p. 117--134, 1995.

[15] K. Schwaber and M. Beedle, *Agile Software Development with Scrum*. Prentice Hall PTR, 2001.

[16] R. Yin, *Case Study Research: Design and Methods, Third Edition, Applied Social Research Methods Series, Vol 5*. Sage Publications, Inc, 2002.

[17] M. Shammi, “Agile Process for Integrated Service Delivery,” Delft University of Technology, 2011. Available: http://resolver.tudelft.nl/uuid:a3eab3e2-29c3-4d19-8208-6f12a7111d8c

Agile Service Development: A Rule-Based Method Engineering Approach[1]

Stijn Hoppenbrouwers[1] Martijn Zoet[2],
Johan Versendaal[2,3], Inge van de Weerd[3],
[1] Radboud University Nijmegen, the Netherlands, {s.hoppenbrouwers@cs.ru.nl}
[2] University of Applied Sciences, Utrecht, the Netherlands,
{martijn.zoet, johan.versendaal}@hu.nl
[3] Utrecht University, Utrecht, the Netherlands,
{j.versendaal, i.vandeweerd, }@cs.uu.nl

Businesses may apply concepts of agility as a strategy to take up challenges in the rapidly changing business environment. Agility is defined as "the ability of a sensitive [organization] that exhibits flexibility to accommodate expected or unexpected changes rapidly, following the shortest time span, using economical, simple and quality instruments in a dynamic environment and applying updated prior knowledge and experience to learn from the internal and external environment" (Qumer en Henderson, 2007). This definition positioned in the context of agile service development asserts that an organization should be able to create or adapt a (business) service efficiently and effectively when changes occur in its environment.

Agile development is not an alien concept in management and information systems research. It plays some role in existing work on *situational method engineering* in software product development literature (Olle et al, 1991; Kumar and Welke, 1992; Brinkkemper, 1996; Van de Weerd et al., 2006). Based on situational factors distilled from the project, meta-methods composed of outlines or more detailed procedures are selected and integrated into a coherent method appropriate for that specific situation (Brinkkemper, 1996). However, 'situational' is not synonymous to 'agile'. For a method to become truly agile, changing situational factors also have to be linked (if required) to 'run time' changes in the method: quick responses to new situational information, and the installation of short feedback loops applying to the method.

Utilizing the perspective of situationality, method fragments can be used to provide some degree of agility with respect to the project at hand. Regarding the assembly of method fragments, our approach follows the configuration process for situational method engineering as proposed by Brinkkemper (1996). However, our approach adds a second dimension of agility in operational execution. Changes in the environment will not always lead to changes in the executed method but can still influence the operational execution of a specific method fragment.

To realize this, we propose a particular operationalization of the method engineering approach and process in terms of the selection process of method

[1] Previously published as: S. Hoppenbrouwers, M. Zoet, J. Versendaal, I. van de Weerd: Agile Service Development –A Rule-Based Method Engineering Approach. In: J. Ralyté, I. Mirbel and R. Deneckère (eds.): Engineering Methods in the Service-Oriented Context; 4th IFIP WG 8.1 Working Conference on Method Engineering, ME 2011, Paris, France, April 20-22, 2011. Proceedings. IFIP Advances in Information and Communication Technology, Volume 351/2011 pp184-189, DOI: 10.1007/978-3-642-19997-4_17

fragments, situational factors and assembly rules. The idea is that participants are given as much freedom as possible within necessary methodical and contextual constraints (minimal specification), and that the ability to respond quickly to desired changes in the method (as indicated by fast feedback) is optimized: increased agility in our approach is supported by defining method fragments in a rule-based, declarative manner. This approach is inspired by principles and practices from (business) rules management, organizational patterns and game design theory.

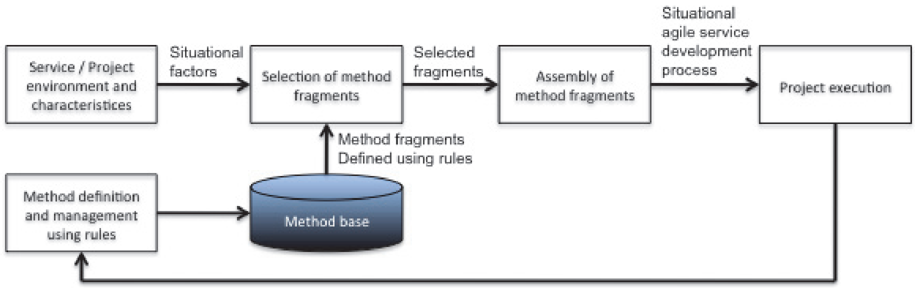


Fig. 1. Method engineering approach for agile service development.

Without claiming that the approach put forward in this position paper will guarantee agility of processes for service development, we believe the approach proposed will allow for considerably better agility than existing practices in ME that are more rooted in imperative style specification of methods and method fragments.

5 References

Brinkkemper, S. (1996). Method engineering: engineering of information systems development methods and tools. *Information and Software Technology*, 38(4), pp 275-280.

Kumar, K and Welke, R.J. (1992). Methodology engineering: a proposal for situation-specific methodology construction. In Cotterman, W.W., Senn J.A. (eds), *Challenges and Strategies for Research in Systems Development*..

Olle, T.W., Hagelstein, J., MacDonald, I.G., Rolland, C., Sol, H.G., van Assche, F.J.M. and Verrijn-Stuart, A.A. (1991). *Information Systems Methodologies: a Framework for Understanding (2nd edition)*. Addison-Wesley.

Qumer, A., Henderson-Sellers, B., (2007). An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Information and Software Technology*, 50(4), pp 280-295.

Weerd, I. van de, Versendaal, J., & Brinkkemper, S. (2006). A product software knowledge infrastructure for situational capability maturation: Vision and case studies in product management. *Proceedings of the 12th Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ'06)*, Luxembourg, pp 97-112.

Towards a Service System Ontology for Service Science

Elisah Lemey and Geert Poels

Center for Service Intelligence
Faculty of Economics and Business Administration
Ghent University
Tweekerkenstraat 2, 9000 Gent, Belgium
{Elisah.Lemey, Geert.Poels}@UGent.be

Service Science is a new interdisciplinary approach to the study, design, implementation, and innovation of service systems. However due to the variety in service research, there is no consensus yet about the theoretical foundation of this domain. In this paper we clarify the service systems worldview proposed by Service Science researchers Spohrer and Kwan by investigating its foundational concepts from the perspective of established service theories and frameworks [1]. By mapping the proposed service system concepts on the selected service theories and frameworks, we investigate their theoretical foundations, examine their proposed definitions, discover their likely relationships, and identify a number of issues that need further discussion. This analysis is visualised in a multi-view conceptual model (in the form of a UML class diagram) which we regard as a first step towards an explicitly and formally defined service system ontology.

The basis for the UML class diagram are the ten foundational concepts of the service systems worldview used by Spohrer and Kwan to explain the diversity and complexity of service systems: *entity*, *resource*, *access right*, *ecology*, *interaction*, *value proposition based interaction*, *governance mechanism based interaction*, *outcome*, *measure*, and *stakeholder*.

The definitions of these concepts are compared with alternative definitions originating in six other service frameworks and theories. We aim to identify which additional concepts from these theories and frameworks can be incorporated in the model to further refine and extend the service systems worldview. By mapping the foundational concepts to the concepts used in traditional service research areas we identify commonalities and differences in interpretation which may help to find a common understanding of the service systems worldview. Also, if we want to create one scientific basis for Service Science research it is crucial that established service frameworks and theories connect to this scientific basis.

Our choice of theories was mainly guided by previous Service Science research. In a joint white paper of IBM and Cambridge University's Institute for Manufacturing the worldview of Service Dominant Logic (SDL) is indicated as a possible theoretical basis for Service Science. Furthermore, other proponents of Service Science propose the Unified Service Theory (UST), the work system method and the service quality gaps model as interesting theories to draw from. As recent Service Science research indicates the need to introduce a system focus in the study of service systems, we also included the system theoretic view of service systems of Mora et al. Finally, we included a service ontology based on the DOLCE upper-level ontology.

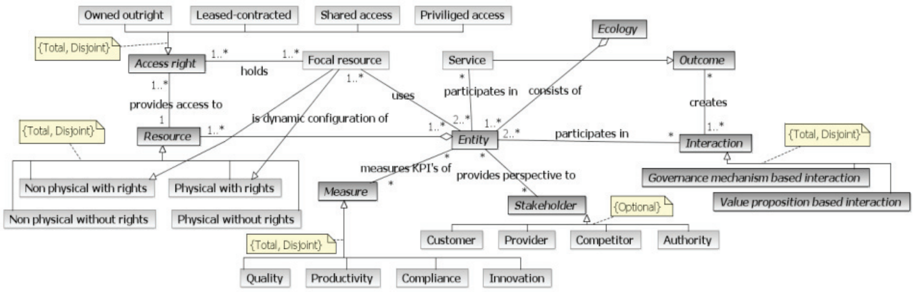


Fig. 1. . UML class diagram of service systems worldview

Our contribution to the emerging research area of Service Science is twofold. First of all, a UML class diagram for the ten foundational concepts is presented (Fig. 1). This diagram is aimed at facilitating the presentation and discussion of the foundational concepts as it also uncovers and shows their relationships. The diagram provides the basis for elaborating a service systems ontology and a meta-model for modelling of service systems. Second, the investigation of the theoretical foundation (if any) and the search for additional concepts which can be marked as foundational, can be seen as a theoretical evaluation of the completeness and relevancy of the set of foundational concepts proposed by Spohrer and Kwan [1]. It provides elements for the further discussion, enhancement, and ultimately (and hopefully) consensual agreement of a service systems conceptualisation for Service Science. Our conceptual analysis points out that more or less all of the foundational concepts and their proposed specialisations are covered by one, many or in some cases even all reviewed service theories or frameworks. We identified a couple of issues that need further discussion and elaboration, e.g., because of conflicting views when mapping foundational concepts to the concepts of different service theories. Overall, however, our analysis shows that there is evidence of theoretical support for the proposed service systems worldview.

An interesting finding is that, although SDL was initially proposed as the philosophical foundation for the service systems worldview, our analysis indicates that the service system conceptualisation put forward by Spohrer and Kwan is developing beyond SDL. The resemblance with the system theoretic approach of Mora et al. shows a shift towards systems thinking which should be further explored in the future.

- The full paper will appear in the proceedings of ICSOC 2011 (Paphos): Elisah Lemey and Geert Poels: Towards a Service System Ontology for Service Science. Full paper accepted for the 9th International Conference on Service Oriented Computing (ICSOC 2011), Paphos, Cyprus, 5-8 December 2011

1. Spohrer, J., Kwan, S.K.: Service Science, Management, Engineering, and Design (SSMED): An Emerging Discipline-Outline & References. International Journal of Information Systems in the Service Sector (IJISSS) 1, 1-31 (2009

Session 3: Business Process Modeling

Business Process Simulation for Management Consultants: A DEVS-Based Simplified Business Process Modelling Library*

Igor Rust[1], Deniz Cetinkaya[1], Mamadou Seck[1], and Ivo Wenzler[2]

[1] Systems Engineering Group; Faculty of Technology, Policy, Management; Delft University of Technology; Jaffalaan 5, 2628BX, Delft, THE NETHERLANDS
[2] Accenture Nederland; Gustav Mahlerplein 90, 1082 MA, Amsterdam, THE NETHERLANDS

Extended abstract

A business process is a series of activities that produces a product or service for a customer. Business Process Modelling (BPM) is the activity resulting in a representation of an organisation's business processes so that they may be analyzed and improved. A distinction can be made between static modelling and dynamic modelling of business processes. Static modelling tools often provide a graphical process representation, for example simple flowcharts, IDEF0 or BPMN diagrams to depict business processes. Business Process Simulation (BPS) tools, provides the possibility to simulate and evaluate the dynamic behaviour of business processes.

The usefulness of business process simulation was proven by many authors and various simulation tools are available, still many management consultants and business analysts rely on simple static process mapping methods. Some reasons for the lack of adoption are that much experience is needed to develop valid simulation models and simulation model development is time consuming and costly. More specifically, there is a lack of business process simulation tools which supports an easy and quick approach of modelling and analysis of business process by consultants and business analysts. This paper presents a business process simulation method to support management consultants to model, simulate and analyze business processes in a well defined manner.

A user-centred design (UCD) approach is chosen to increase the likelihood that a designed and developed solution is found usable by its end-users, the management consultants. User-centred design is concerned with incorporating the end-users perspective during the design and development process to achieve a usable system. To incorporate management consultants in this research, a series of design and evaluation rounds are held (in the form of workshops) with management consultants of a large international management consultancy firm.

* Original version of this paper: Rust, I., Cetinkaya, D., Seck, M., Wenzler, I. (2011). Business Process Simulation for Management Consultants: A DEVS-Based Simplified Business Process Modelling Library. In Proceedings of The European Modelling & Simulation Symposium 2011. Genova : Universita di Genova, 2011. 709-714.

The modelling representation that was the outcome of the design research process is based on the Business Process Modelling Notation (BPMN). BPMN is an industry-wide standard for modelling of business processes. A set of business process modelling (BPM) elements are determined, which allow modelling of business processes by consultants as how they actually perceive business processes.

We use the Discrete Event System Specification (DEVS) to specify our simulation models. The suggested BPM elements and some supplementary elements are matched to DEVS simulation components. For every element a state-diagram was developed and validated. The supplementary components are developed to support some of the needed simulation functionality as discussed by the consultants.

DSOL, which stands for "Distributed Simulation Object Library", was selected to provide the simulation and execution functionalities. DSOL is a proven multi-formalism simulation library which can be considered as a generic purpose simulation tool. It is written in the Java programming language and has been used effectively in various simulation projects. DSOL also supports execution of simulation models based on the DEVS formalism through the DEVSDSOL library (which is compatible with hierarchical DEVS). Each DEVS component has been implemented in Java and these components are executable with DEVSDSOL simulation library.

In order to provide a higher level abstraction mechanism to our library, we applied the model-driven development framework presented by Cetinkaya (2011). MDD4MS is a model-driven development framework for modelling and simulation. The framework suggests an M&S life cycle with five stages (Problem Definition, Conceptual Modelling, Specification, Implementation and Experimentation), metamodel definitions for different stages, model to model (M2M) and model to text (M2T) transformations for the metamodels and a tool architecture for the overall process. MDD4MS presents a sample prototype implementation which is developed in Eclipse.

In this study, we used the BPMN editor to draw our business process models. Since the MDD4MS prototype provides generic model transformation rules for BPMN, we rewrote some rules for BPMN2DEVS M2M transformation. In this way, we directly transformed the visual modelling elements to the DEVS components that we implemented in our library. Once we have the DEVS model, the MDD4MS prototype automatically generates the DEVSDSOL model and the java code for coupled components that uses the implemented classes for BPM modeling elements in our library. In other words, visual business process models, drawn by the BPMN editor, are transformed to executable Java code and they can be simulated with DSOL.

This work proposed a new modelling approach for management consultants to model and analyse business processes based on a proven theory, industry-wide standards and active end-user involvement during the design process. A library of DEVS-based BPMN modelling elements is implemented with Java that uses the DSOL simulation library to provide the simulation capabilities.

Merging Computer Log Files for Process Mining: an Artificial Immune System Technique

Jan Claes and Geert Poels

Department of Management Information Systems and Operations Management
Faculty of Economics and Business Administration
Ghent University, Tweekerkenstraat 2, 9000 Ghent, Belgium
{jan.claes,geert.poels}@ugent.be

Abstract. Process mining techniques try to discover and analyse business processes from recorded process data. These data have to be structured in so called *computer log files*. If processes are supported by different computer systems, merging the recorded data into one log file can be challenging. In this paper we present a computational algorithm, based on the Artificial Immune System algorithm, that we developed to automatically merge separate log files into one log file. We also describe our implementation of this technique, a proof of concept application and a real life test case with promising results.

Keywords: Business Process Modelling, Process Mining, Log File Merging

This paper has been published as: Claes, J., Poels, G.: Merging Computer Log Files for Process Mining: an Artificial Immune System Technique. BPM 2011 Workshops, LNBIP. Springer (2011).

1 Introduction

Process mining techniques are used to discover and analyse business processes in a semi-automatic way. Starting from all kinds of recorded process data (called *log files*) process mining tries to automatically discover the structure and properties of the business processes, which can be visualised in business process models.

Three actions have to be taken before process discovery and analysis techniques can be performed: searching for data in the IT support systems, structuring these data (i.e. identifying single process steps (events) and groups of process steps that belong to the same process execution (process instances)), and converting these data to the format required by the process mining tool. If process data are found in different sources, then a fourth action is required: merging the data into one computer log file.

In this paper we present an automated technique for merging already collected, structured and converted process data according to an Artificial Immune System (AIS) algorithm, which is based on the features and behaviour of the vertebrate immune system. By automating this fourth action of the preparation step, we try to broaden the benefits of process mining to an extended part of the overall process mining procedure, because the automation makes the merge step in the preparation phase faster (speed), the use of data from multiple systems is facilitated (completeness) and the way these data are merged is less subjective than when performed manually (correctness).

Our approach is implemented in ProM, a well known academic process mining tool, which implies that for our implementation we assume the different data sets are first separately structured and converted to the ProM file format. Fig. 1 shows the steps for our solution implementation.

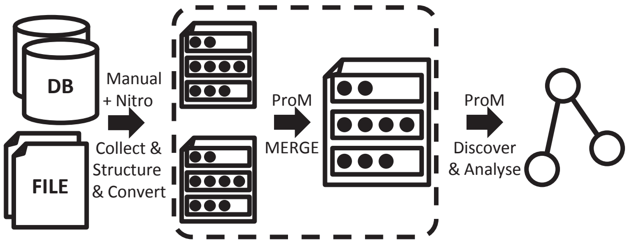


Fig. 1. Merging data of different sources can be performed after structuring and converting to a tool-specific file format. We implemented our merge technique in the ProM analysis tool itself.

2 Technique

The merging of two computer log files consists of two steps: (i) linking together traces of both logs that belong to the same process execution and (ii) merging these traces into one trace to be stored in a new log file. We assume reliable and comparable timestamps are available in the original logs causing the second step to be a simple exercise of chronological ordering of all the events of linked traces into one new trace in the resulting merged log file.

In our opinion, more than one factor can indicate that two traces should be linked We looked for existing techniques that incorporate multiple indicators in their solution procedure and found our inspiration in the Artificial Immune System algorithm. This algorithm uses an affinity score throughout the entire calculation procedure where this score points to the best solution. We used a combined indicators function to derive the affinity score for the algorithm. Each scored solution is not more than a set of linked traces between the two log files. By combining our different assumed indicators, a solution with a high score has a higher chance to be an optimal solution because most combined indicator value points to that solution.

3 Experiment results

We have tested our technique with a simulated example and with a real case example. The benefit of using simulation is that the correct solution (i.e. the process to be discovered) is known end that properties like time difference or noise can be controlled.

The results of both series of tests revealed a nearly perfect merge if both logs used the same trace identifiers. With different trace identifiers in both logs in all cases the correct *number* of links was found, but when traces run partly in parallel, there seems to be too little information left to find the right links. The amount of noise in the logs seems to have little impact on the correctness of the identified links

The full published article can be found at http://processmining.ugent.be/post.php?post=pubbpi2011

Questioning the Design of Business Process Maturity Models

Amy Van Looy[1,2], Manu De Backer[1,2,3,4], Geert Poels[2]

[1] University College Ghent, Department of Management & Informatics,
Voskenslaan 270, B-9000 Ghent, Belgium
{amy.vanlooy, manu.debacker}@hogent.be
[2] Ghent University, Department of Management Information Science & Operations Management, Tweekerkenstraat 2, B-9000 Ghent, Belgium
{Amy.VanLooy, Manu.DeBacker, Geert.Poels}@UGent.be
[3] University of Antwerp, Department of Management Information Systems,
Prinsstraat 13, B-2000 Antwerp, Belgium
{Manu.DeBacker}@ua.ac.be
[4] K.U.Leuven, Department of Management Informatics,
Naamsestraat 69, B-3000 Leuven, Belgium
{Manu.DeBacker}@econ.kuleuven.be

Abstract. The importance of business process management goes without saying. As its realization is less straightforward, maturity models have been developed to gradually assess and improve business processes. Although their aim is to assist organizations, the proliferation of maturity models also confuses organizations. They have no overview of existing models and their differences, which makes an informed choice difficult. Choosing the right business process maturity model (BPMM) is however important, as previous research indicated the existence of different maturity types being measured by the existing models [1]. We now add further design elements to our comparative framework by conducting a content analysis of 69 BPMMs. Afterwards, the identified design elements are transformed into a questionnaire that practitioners can use to find the BPMM that best fits their needs. In this paper, we present 16 questions to be included in the questionnaire, without elaborating on the mapping of individual maturity models.

Keywords: business process maturity, business process management, business process orientation

1 Introduction

Today's globalized market is characterized by demanding customers and growing IT possibilities. Organizations are therefore increasingly relying on their way of working, i.e. business processes, to excel [2,3]. However, merely modeling and deploying a business process does not imply that your business process is also an excellent one, or at least a good one. Therefore, the notion of 'maturity' is introduced as a measure to indicate how excellent business processes can perform [2,4]. Maturity requires continuous process improvements, which are not easy to realize. Hence, business process maturity models (BPMMs) have been designed from which organizations gradually benefit. In general, a maturity model (MM) is a tool to systematically assess and improve capabilities, i.e. abilities or competences, to reach a goal. Translated to BPMM, it concerns the capabilities of business processes and their organizations to reach business (process) excellence. An example is given below.

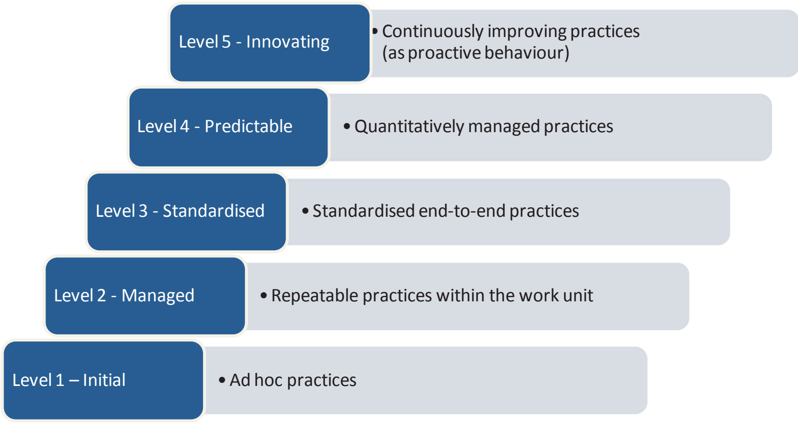


Fig. 1. A BPMM example [5]

Currently, a BPMM proliferation exists [6]. This proliferation raises questions about the differences between BPMM designs. To our knowledge, some comparative attempts have been made by Hüffner [7], Lee, Lee, and Kang [8], Maier, Moultrie, and Clarkson [9], and Rosemann and de Bruin [10]. Nonetheless, they do not intend to offer a comprehensive comparative study on a large number of BPMMs. For this purpose, two research questions are raised.

RQ1. On which design elements do existing BPMMs differ?
- ⇨ A comparative framework is built to classify existing BPMMs.

RQ2. Which BPMM must be chosen when?
- ⇨ A questionnaire is derived from the comparative framework to obtain a practical instrument that managers can use while choosing a BPMM.

We start with defining a maturity model in section 2, and more specifically a BPMM in section 3. Section 4 clarifies the methodology. It is followed by presenting (section 5) and discussing (section 6) the BPMM comparative framework (RQ1) and the BPMM questionnaire (RQ2). Finally, we summarize the results and future research.

2 Maturity Model (MM)

An overall definition for a maturity model (MM) is provided by Tapia *et al.* [11]: '*MMs have been developed to assess specific areas against a norm. Based on maturity assessments, organizations know the extent to which activities in such areas are predictable*' [11,pp.71].

MMs share some design elements, independent of whether they deal with business processes, business-IT alignment, e-government, quality management, etc. Table 1 lists those design elements found in the literature on MM design [12,13,4,14,11,15]. The emphasis is on *who* measures maturity (i.e. assessors – 'WHO'), and *how* it is measured (i.e. assessment method – 'HOW'). Furthermore, the table clarifies *what* is measured as maturity, i.e. capability areas and their improvements necessary to reach each consecutive level (i.e. improvement method – 'WHAT').

Table 1. The MM design elements.

	[12]	[13]	[4]	[14]	[11]	[15]
Assessors – WHO						
• Assessment unit	X	X	X	X	X	X
• Lead assessor	X	X	X	-	X	X
• Other assessors and respondents	X	X	X	X	X	X
Assessment method – HOW						
• Data collection technique to obtain information to assess	X	X	X	X	-	X
• Calculation to interpret the collected data as lifecycle levels	X	X	X	-	X	X
• Representation to visualize lifecycle levels	-	X	X	X	X	X
Improvement method – WHAT						
• Capability areas to assess and improve	X	X	X	X	X	X
• Lifecycle levels	X	X	X	X	X	X
• Architecture or road map, to link capability areas with levels	X	X	X	X	X	X

3 Business Process Maturity Model (BPMM)

Translated towards business processes, BPMMs are evolutionary models for measuring (AS-IS) and improving (TO-BE) maturity, or '*the extent to which an organization consistently implements processes within a defined scope that contributes to the achievement of its business goals*' [16,pp.2]. Mature business processes acquire the necessary capabilities to reach excellence. Capabilities are competencies (e.g. skills and knowledge) to achieve the targeted results, i.e. the ability to perform, or the expected performance of a business process. Related capabilities are collected into capability areas. Maturity levels indicate the growth through all capability areas together. Sometimes, capability levels are present to indicate the growth through each capability area separately [17,18,10].

Capability areas differentiate a BPMM from other MMs. Previous research [1,19] has identified six main capability areas from the definitions for three fundamental domains in the business process literature: (1) business process (BP), (2) business process management (BPM), and (3) business process orientation (BPO).

First, business process definitions implicitly focus on *business process modeling* and *deployment*. The latter means running processes in real life. It requires modeling or predefining business processes in textual or graphical descriptions [20]. For instance, *'a process is a series of interconnected activities that takes input, adds value to it, and produces output. It's how organizations work their day-to-day routines'* [2,pp.xxii]. Both aspects are selected as main capability areas.

Secondly, BPM involves continuously managing and improving business processes, guided by process owners. Gillot [21], Gulledge Jr. and Sommer [22] summarize four foci in BPM definitions: (1) *modeling*, (2) *deployment*, (3) *optimization*, or improving business processes based on real metrics, and (4) the *management* of business processes, each with a process owner and a cross-functional team. For instance, Weske [20] defines BPM as '*concepts, methods, and techniques to support the* (1) *design,* (4) *administration,* (2) *configuration, enactment, and* (3) *analysis of business processes*' [20,pp.5]. Similarly to BP, these four foci are selected as main capability areas. BPM differs by also addressing optimization and managerial efforts for one, more or all business processes.

Some authors go beyond these four BPM areas by also referring to organization management. Particularly, by adopting (5) a process-oriented culture with rewards linked to the performance of business processes instead of departments, and (6) a horizontal structure [23]. For instance, McCormack [24] defines BPO as '*an organization that emphasizes process, a process oriented way of thinking, outcomes and customers as opposed to hierarchies*'[24,pp.6]. Although the distinction between BPM and BPO is not always explicitly made, e.g. in [10], it allow separately examining the different nuances.

Consequently, six main capability areas are derived from the BP, BPM and BPO definitions. Each area must be assessed and improved in order to reach business process maturity. It turned out that some BPMMs measure BPM maturity, by addressing the first four capability areas, whereas others measure BPO maturity, by also addressing the cultural and structural capability areas [1].

4 Methodology

4.1 BPMM Sample (N=69)

The research scope was set to generic business processes. It excludes BPMMs addressing specific process types, such as in the initial software engineering maturity models. However, models that integrate various specific BPMMs were withheld to represent those specific topics. Also maturity models for supply chains and collaboration processes were selected to study cross-organizational value chains.

Data was collected during the second quarter of 2010. First, we searched for articles in academic databases and search engines on the Internet by using the combined keywords of '*process*' and '*maturity*'. Secondly, we traced the references in the identified articles to get access to other relevant sources.

We acknowledge some restrictions regarding the accessibility of articles (in Ghent University engines), the language (English, Dutch, French or German), and the keywords. Notwithstanding these limitations, the technique turned out to be fruitful in terms of the number of maturity models identified.

4.2 Content Analysis

Due to the lack of a Meta theory on BPMM designs, the variables within each design element of Table 1 were primarily identified by the 'Grounded Theory' [25], which systematically generates: (1) 'codes', i.e. BPMM attributes or variable values, (2) 'concepts', i.e. variables, (3) 'categories', i.e. design elements to group variables, and (4) a 'theory', i.e. a comparative framework. The successive coding stages were:

- initial (open) coding: we read the collected texts by constantly going back and forth to compare existing BPMM designs. Hence, we identified possible attributes and variables;
- intermediate (axial) coding: the attributes and variables were rethought and linked to the initial design elements. It resulted in the variables to be included in the framework;
- advanced (selective) coding: we reread the collected texts to encode what is literally written in these texts to the obtained variables.

5 Results

5.1 BPMM Comparative Framework (RQ1)

The comparative framework, shown in Figure 2, was built iteratively.

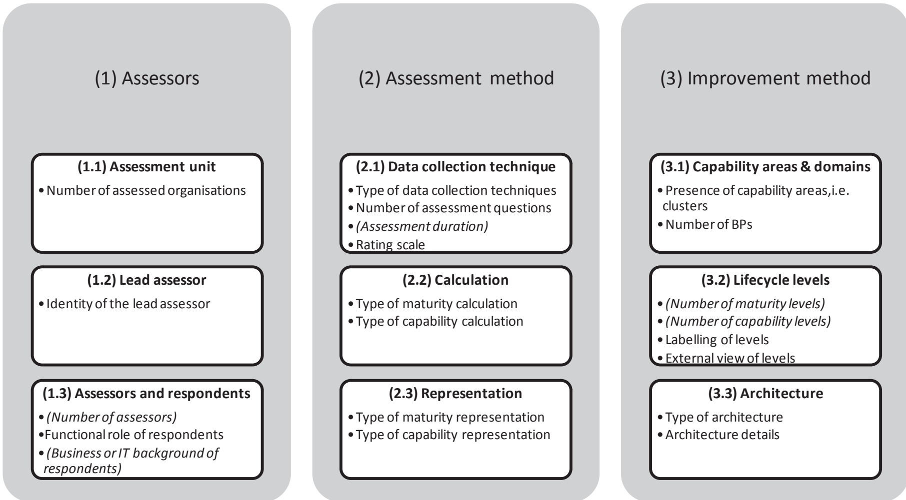


Fig. 2. The comparative BPMM framework.

After coding the identified variables, descriptive statistics were used to enhance our dataset and to keep only those variables important to our questionnaire (RQ2). Variables without fundamental differences among the collected BPMMs were eliminated, because of less differentiating power. This applies to the number of lifecycle levels, which was mostly three to six levels, with a mode of five levels. To maintain the quality of the questionnaire, we also decided to eliminate all variables with missing values on more than one third of the collected BPMMs, i.e. (1) the number of assessors, (2) the background of respondents, and (3) the assessment duration. All variables eliminated at this stage are italicized in Figure 2. Nonetheless, they remain important design elements.

5.2 BPMM Questionnaire (RQ2)

The final step is to transform the comparative framework into a questionnaire that practitioners can use to select a BPMM. For this purpose, the 16 resulting variables were reformulated into a similar number of questions, available in appendix. Their comprehensiveness was approved by other BPM scholars within the faculty.

6 Discussion

We have found 16 design elements on which existing BPMMs substantially differ. Consequently, they can be used to motivate the choice for one or another BPMM. We deliberately excluded a discussion on the methodology and validation used to design the particular BPMMs. Notwithstanding their relevance for BPMM credibility, such information remains elusive in many design documents (e.g. found on websites or white papers). However, this does not necessarily exclude rigorous research, which makes a comparison arguable. For reasons of objectivity, the comparative framework was restricted to the BPMM design itself.

We advise organizations to choose a BPMM that best fits their needs. Therefore, our questionnaire allows answering only those questions that are considered as relevant by a particular organization. However, given the importance of capability areas, we make Q11 mandatory. This implies that an organization must decide whether to address BPM maturity, or BPO maturity. Next, further refinements can be optionally made by answering the other 15 questions.

7 Future Work

The 16 questions will be used to create a decision table, which visually maps only the proven BPMMs to the variables and the trade-offs of each variable (i.e. expected efforts and benefits). Its use will be tested in real business scenarios by conducting field studies. We will first ask practitioners to indicate which questions they consider the most important for their organization. For instance, some organizations may prefer a BPMM that certifies the assessed maturity level, whereas other organizations may look for an informal and quick assessment with only a few assessment items. Based on these answers, the decision table will select a BPMM that best fits such requirements. Afterwards, interviews will be conducted to evaluate whether practitioners are satisfied with the result, and whether they will use the resulting BPMM in their organization.

8 Conclusion

Business process maturity has received a lot of attention in the business process literature, but mainly as individual maturity models. To our knowledge, no comprehensive overview currently exists. Our research tries to fill this gap by conducting a comparative study on a sample of 69 BPMMs. This paper only focuses on a small, though important part of that research. Particularly, it presents a questionnaire with 16 questions, derived by a content analysis of the design documents from the sampled BPMMs. It can be used by practitioners to select a BPMM that best fits their organizational needs.

Indeed, organizations wishing to start improving business process maturity must first choose a BPMM out of a wide array. Since existing BPMMs vary on many design elements, this choice may impact their further progression in business process management. Frequently, such organizations are not aware of those differences. Therefore, our questionnaire supports their BPMM choice by considering the most important design differences among existing BPMMs.

References

1. Van Looy, A., De Backer, M., Poels, G.: Which maturity is being measured? A classification of business process maturity models. In: 5th SIKS/BENAIS Conference on Enterprise Information Systems, pp. 7--16, Eindhoven (2010)
2. Harrington, H.J.: Process Management Excellence. Paton Press, California (2006)
3. vom Brocke, J., Rosemann, M.: Foreword. In: vom Brocke, J., Rosemann, M. (eds.): Handbook on Business Process Management 2, pp. vii--ix. Springer, Berlin (2010).
4. Maier, A.M., Moultrie, J., Clarkson, P.J.: Developing Maturity Grids for Assessing Organizational Capabilities: Practitioner Guidance. In: 4th International Conference on Management Consulting, pp. 1--29, Vienna (2008)
5. OMG: Business Process Maturity Model (BPMM) - Version 1.0 (1 June 2008), Online available, `http://www.omg.org/spec/BPMM/1.0/PDF`
6. Sheard, S.A.: Evolution of the Frameworks Quagmire. IEEEComputer 34(7), 96--98 (2001)
7. Hüffner, T.: The BPM Maturity Model - Towards a Framework for Assessing the Business Process Management Maturity of Organizations. GRIN, Munich (2004)
8. Lee, J., Lee, D., Kang, S.: An Overview of the Business Process Maturity Model (BPMM). In: Advances in Web and Network Technologies, and Information Management, pp. 279--288. Springer: Heidelberg (2007)
9. Maier, A.M., Moultrie, J., Clarkson, P.J.: A Review of Maturity Grid based Approaches to Assessing Organizational Capabilities. In: Academy of Management Annual Meeting. California (2008)
10. Rosemann, M., de Bruin, T.: Application of a Holistic Model for Determining BPM Maturity. BPTrends (February 2005)
11. Tapia, R. *et al.*: Towards a Business-IT Alignment Maturity Model for Collaborative Networked Organizations. In: 12th International EDOC conference, pp. 70--81, Munich (2008)
12. Becker, J., Knackstedt, R., Pöppelbuss, J.: Developing Maturity Models for IT Management - A Procedure Model and its Application. Business & Information Systems Engineering 1, pp. 213--222 (2009)
13. de Bruin, T. *et al.* : Understanding the Main Phases of developing a Maturity Assessment Model. In: 16th Australasian Conference on Information Systems, pp. 1--10 (2005)
14. Mettler, T., Rohner, P.: Situational Maturity Models as Instrumental Artifacts for Organizational Design. In: 4th International Conference on DESRIST, pp. 1--9 (2009)
15. van Steenbergen, M. *et al.*: The Design of Focus Area Maturity Models. In: 5th International Conference on DESRIST, pp. 317--332 (2010)
16. ISO/IEC: Information Technology - Process Assessment - Part 7: Assessment of Organizational Maturity - ISO/IEC TR 15504-7:2008(E). ISO/IEC, Geneva (2008)
17. Hammer, M.: The Process Audit. Harvard Business Review (April), 111--123 (2007)
18. McCormack, K.: Supply Chain Management Maturity. In: McCormack, K. (ed.) Business Process Maturity. Theory and Application, pp. 73--103. Booksurge Publishing, South Carolina (2007)
19. Van Looy, A., De Backer, M., Poels, G.: Defining Business Process Maturity. A Journey towards Excellence. FEB Working Paper 2011/725. Ghent University, Ghent (2011)
20. Weske, M.: Business Process Management. Springer, Berlin (2007)

21. Gillot, J.-N.: The Complete Guide to Business Process Management. Booksurge Publishing, South Carolina (2008)
22. Gulledge Jr, T.R., Sommer, R.A.: Business Process Management: Public Sector Implications. Business Process Management Journal 8(4), 364--376 (2002)
23. McCormack, K., Johnson, W.C.: Business Process Orientation: Gaining the e-Business Competitive Advantage. St. Lucie Press, Florida (2001)
24. McCormack, K.: Introduction to the Theory of Business Process Orientation. In: McCormack, K. (ed.) Business Process Maturity. Theory and Application, pp. 1--18. Booksurge Publishing, South Carolina (2007)
25. Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory: Strategies for Qualitative Research. Transaction Publishers, New Jersey (1967)

Appendix

Variables	Questions
(1) Assessors	
(1.1) Assessment unit	
Q1. How many organizations must be included in the assessment? • One, i.e. the maturity model mentions a single organization • More, i.e. the maturity model mentions more organizations	
(1.2) Lead assessor	
Q2. Must the assessment be lead by an independent person? More options are possible. • No • Yes, without certification of the assessment results • Yes, with certification of the assessment results	
(1.3) Assessors and respondents	
Q3. Must people from outside the assessed organization(s) be included as respondents? • No • Yes	
(2) Assessment method	
(2.1) Data collection technique	
Q4. How must information be collected? More options are possible. • Objectively, e.g. by document reviews. • Subjectively, e.g. by questionnaires, interviews, observations.	
Q5. How many questions must be maximally answered in a particular assessment? • 1-20, i.e. twenty questions or less • 21-50, i.e. between twenty-one and fifty questions • >=51, i.e. more than fifty questions	
Q6. Which type of data must be collected? More options are possible. • Qualitative, i.e. with open questions or with nominal or ordinal rating scales • Quantitative, i.e. with discrete, interval or ratio rating scales	
(2.2) Calculation	
Q7. If maturity levels are applicable (Q16: staged architecture), must the resulting maturity level be directly observable (e.g. the exact or lowest score on assessment questions), or indirectly (i.e. requiring calculations or statistical formula)? More options are possible. • Directly • Indirectly	
Q8. Idem Q7, but for capability levels (applicable if Q16: continuous architecture)	

Variables	Questions
(2.3) Representation	
Q9. If maturity levels are applicable (Q16: staged architecture), how must the calculated maturity level be displayed? More options are possible. • Textually (e.g. 'defined' or 'quantitatively managed') • Numerically (e.g. 3, or 3.7, or 67%) • Graphically • Matrix,i.e.table with questions in the rows, levels in the columns, explanations in the cells.	
• Q10. Idem Q9, but for capability levels (applicable if Q16: continuous architecture)	
(3) Improvement method	
(3.1) Capability areas and domains	
Q11. Which capability areas must be primarily assessed and improved? • BPM maturity, i.e. primarily focusing on business process modeling, deployment, optimization and management (e.g. for team initiatives) • BPO maturity, i.e. combining BPM maturity with a process-oriented culture and structure (e.g. for top management initiatives)	
Q12. How many business processes must be assessed and improved? More options are possible. • One, i.e. a single business process or sub process • More, i.e. more than one, but not all business processes. Assessment questions deal with a particular business domain or value chain and their (sub) processes • All, i.e. all business processes in the involved organization(s) or supply chain. Assessment questions focus on how the organizations deal with business processes in general	
(3.2) Lifecycle levels	
Q13. What must the labels of the lifecycles indicate? More options are possible. • Business process optimization E.g. from 'initial', to 'managed', 'standardized', 'predictable', and 'innovating' processes • Business process management E.g. from 'BPM initiation', to 'BPM evolution', and 'BPM mastery' • Business process integration E.g. from 'ad hoc', to 'defined', 'linked', 'integrated', and 'extended' processes	
Q14. To which extent must the lifecycles take into account possible relationships between individual organizations? • No notion, i.e. all lifecycle levels are limited to one organization E.g. from 'initial', to 'managed', 'standardized', 'predictable', and 'innovating' processes • Highest levels, i.e. as from the highest levels, external relationships are taken into account E.g. from 'ad hoc', to 'defined', 'linked', 'integrated', and 'extended' processes • All levels, i.e. as from the lowest levels, external relationships are taken into account E.g. from 'ad hoc', to 'planned', 'aware', and 'reflexive' collaboration	
(3.3) Architecture	
Q15. Must a road map be defined per capability area and/or overall maturity? More options are possible. • Continuous, i.e. capability levels exist and are linked to each capability area separately. • Staged, i.e. maturity levels exist and are linked to all capability areas together.	
Q16. How much guidance must the road map give on your journey towards higher maturity levels and/or capability levels? • Descriptive, i.e. the road map is limited to a high-level description, without criteria. • Implicit prescriptive, i.e. the road map has criteria interwoven in the assessment questions • Explicit prescriptive, i.e. the road map has a separate list of criteria	

Business Service Modeling for the Service-Oriented Enterprise [1](Extended Abstract)

Jeewanie Jayasinghe Arachchige, Hans Weigand, Manfred Jeusfeld

Tilburg University, P.O.Box 90153,
5000 LE Tilburg, The Netherlands
{J.JayasingheArachchig, H.Weigand, Manfred.Jeusfeld}@uvt.nl

Keywords: Business Services, REA, Service-Oriented Design

Service-oriented architectures are the upcoming business standard for realizing enterprise information systems, thus creating a need for analysis and design methods that are truly service-oriented. Most research on this topic so far takes a strict software engineering perspective. For a proper alignment between the business and the IT, a service perspective at the business level is needed as well. Although few researchers realized the importance of business thinking at service design, notation for service design is still lacking.

Using an MDA approach, [1] introduces a new business service and resource modeling language - BSRM based on the Resource-Event-Agent (REA) business ontology. The constructs of the BSRM language and their relationships are grounded in a meta-model which provides comprehensive specification using UML notation. Following the REA, we defined the service as a specialization of economic resource in the meta-model. As the resources are first class citizen of REA, BSRM gives a better insight to the value co-creation which is the main focus of services. Further, we distinguish two service specializations: exchange service and conversion service, corresponding to the two basic REA dualities. Each of them corresponds to a group of decrement and increment economic events in REA. The service classification model developed in [2], categorizes services into several categories. Among these service categories enhancing services and the sub-services which have part of relationship to the core services are special types of services that play a major role in service modeling at business level. These service categories correspond to different roles of the service type in our model.

The concept of enhancing services which adds value to the any other service, is introduced as another service category in our model. Considering the situation where core-service realization involves multiple value activities and it makes sense to view these value activities as independent services that are shared by different contexts, we identified the forth category of services as sub-services. Sub- service can be a part-of

[1] Jayasinghe Arachchige J., Weigand, H., and Jeusfeld, M.: Business Service Modeling for the Service-Oriented Enterprise. International Journal of Information System Modeling and Design, Vol. 3(1) - 2012, (accepted).

service of core service which is called core sub-service, or enhancing sub-service which is called coordination service. [3] extended the basic REA stockflow concept by adding specialization to the stockflow relationship as inflow and outflow. We adopt this specialization into our model as a relationship between Economic Resource Type and Service Type.

All these concepts which are defined in the meta-model, are denoted with a simple modeling notation called -BSRM in [1]. A major difference of BSRM with other service modeling approaches is the resource perspective of services. We have included resources not only because they are needed to describe service effects, but also since resources play a prominent role in new service design [4]. Secondly, in contrast to most other approaches that only consider one type of relationship between services, we identify many service linkages, in order to catch more semantics, keeping in mind the tenet that meaning is captured in structural relationships. The BSRM design steps provide a simple direction to a designer to use this modeling language at CIM level in any real situation.

As it is not feasible to grasp all relevant concerns to a single model, we mapped the service modeling language with complimentary models, in particular value network (e3value), data model (ER) and process models (BPMN). By mapping the meta-model with other model types at CIM and PIM level, we have shown how it can support truly service-oriented IS development.

BSRM is not only guided by meta-model, it has been formalized using the meta-modeling facilities of Conceptbase[5]. This does not only ensure formal validation but has also provided us immediately with a workable BSRM modeling tool. Further, we evaluated the viability of BSRM by feature comparison and applying to a real world case study in the logistic domain.

BSRM is not only a vehicle for communication with business analysts but also a basis for service-oriented value analysis. All in all, we have described and evaluated BSRM that we claim to be the first specific service modeling language at CIM level.

References

1. Jayasinghe Arachchige J., Weigand, H., and Jeusfeld, M. (2012): Business Service Modeling for the Service-Oriented Enterprise. International Journal of Information System Modeling and Design, Vol. 3(1), (accepted).
2. Weigand H., Johannesson, P., Andersson, B., & Bergholtz , M. (2009). Value-based Service Modeling and Design: Toward a Unified View of Services. In Proc. of CAiSE '09, Springer, 410-424.
3. Gailly, F., Laurier, W., Poels, G. (2008). Positioning and Formalizing the REA enterprise ontology, International Journal of Information Systems 22, 219-248.
4. Froehle, C.M., Roth, A.V. (2007). A Resource-Process Framework of New Service Development, Production and Operations Management, 16(2), 169-181,183-188.
5. Jeusfeld, M.A., Jarke, M., & Mylopoulos, J. (2009). Metamodeling for Method Engineering. MIT Press.

Session 4: Collaborative Modeling

Assessing Collaborative Modeling Quality Based on Modeling Artifacts*

D. (Denis) Ssebuggwawo[1], S.J.B.A. (Stijn) Hoppenbrouwers[1], and H.A. (Erik) Proper[1,2]

[1] Institute of Computing and Information Sciences, Radboud University Nijmegen
Heyendaalseweg 135, 6525 AJ Nijmegen, The Netherlands, EU.
D.Ssebuggwawo@science.ru.nl, stijnh@cs.ru.nl
[2] Public Research Centre – Henri Tudor, Luxembourg, EU.
erik.proper@tudor.lu

Abstract. Collaborative modeling uses and produces modeling artifacts whose quality can help us gauge the effectiveness and efficiency of the modeling process. Such artifacts include the modeling language, the modeling procedure, the products and the support tool or medium. To effectively assess the quality of any collaborative modeling process, the (inter-) dependencies of these artifacts and their effect on modeling process quality need to be analyzed. Although a number of research studies have assessed and measured the quality of collaborative processes, no formal (causal) model has been developed to assess the quality of the collaborative modeling process through a combination of modeling artifacts. This paper develops a Collaborative Modeling Process Quality (CMPQ) construct for assessing the quality of collaborative modeling. A modeling session involving 107 students was used to validate and measure the quality constructs in the model.

Key words: Collaborative Modeling, Modeling Process Quality, Modeling Artifacts, Instrument Validation, Structural Equation Modeling

1 Introduction

While a number of approaches have been developed to measure and evaluate the quality of a collaborative modeling process, e.g. its successfulness [1] and users' satisfaction [2], there has not been any study that integrates the assessment of various modeling artifacts to determine the quality of a collaborative modeling process. Driven by the need to determine the efficiency and effectiveness of the modeling process, we propose an evaluation method that indeed integrates the assessments as an alternative method for determining the quality and successfulness of, and users' satisfaction with, a modeling process.

* This paper is published in: P. van Bommel et al. (Eds.), PoEM 2010, LNBIP 68, pp. 76-90. Springer-Verlag.

2 CMPQ Construct Measurement Instrument

Our assessment approach evaluates quality of the collaborative effort through the quality of the modeling artifacts, used in and produced during collaborative modeling. Specifically, we look at the following constructs: Perceived Quality of the Modeling language (PQML), Perceived Use of the Modeling Procedure (PUMP), Perceived Quality of the Modeling Product (PQEP) and Ease of Use of the Medium (Support tool) (EOUM) to develop an integrated approach and a Collaborative Modeling Process Quality Assessment (CMPQ) construct for assessing the quality of the collaborative modeling process. Secondly, we wanted to measure the (inter)dependencies (causation, correlations, etc.) of the modeling artifacts on each other and their effect and impact on the overall quality of the modeling process. To this effect, we apply Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) techniques on a conceptual model (Model 1) and a competing model (Model 2).

3 Major Findings and Conclusion

The first observation about the results of the confirmatory analysis (CFA is that the (standardized) factor loadings of the the conceptual model (Model 1) and the competing model (Model 2) are close. In fact they are the same for the PQML and PUMP constructs while slight differences are noticed for the PQEP and EOUM. This closeness of the results indicates that the Model used in the EFA was a good conceptual model. To determine the possibility of Model 2 being preferred to Model 1, we compare the model fit indices of both models to determine which ones are near or better than the threshold values (see [3] for these threshold values). Through the two known statistical techniques: Exploratory Factor Analysis (EFA) and Confirmatory Factor Analysis (CFA) we can confirm that the approach is sound and the research instrument passes the validity and reliability tests. The contribution of this paper is thus two-fold. First, it develops a method of assessing collaborative modeling quality based on modeling artifacts used in, and developed during the collaborative modeling effort. Second, a validated instrument for measuring the developed constructs and assessing the quality of the CMPQ construct is presented and was properly validated.

References

1. Davison, R.: An Instrument for Measuring Meeting Success: Revalidation and Modification. Information and Management 36, 321–328 (1999)
2. Briggs, R.O., Reinig, B.A., Vreede, G.J. de: Meeting Satisfaction for Technology Supported Groups: An Empirical Validation of a Goal-Attainment Model. Small Group Research 37, (2006)
3. Straub, D., Boudreau, M.-C., Gefen, D.: Validation Guidelines for IS Positivist Research. Communication of the AIS 13, 380–427 (2004)

A Context-aware Inter-organizational Collaboration Model Applied to International Trade[1]

Jie Jiang, Virginia Dignum, Yao-Hua Tan, Sietse Overbeek

Delft University of Technology, 2628 BX Delft, the Netherlands

1 Introduction

With the development of international trade, collaborative organizations are involved in the value chain to accomplish their cooperative goals. Business organizations try to operate as efficiently as possible while governmental organizations have to regulate business performance. Governmental regulation of multi-organization alliances is not only complex but also time-consuming. Hence, interactions between business and governmental organizations is changing from monolithic control by regulatory authorities to distributed environments where private enterprises are free to regulate their affairs within the boundaries set by the regulatory authorities. The former way of controlling is called *direct control* and the latter is named *self-regulation*. In order to determine the effects and possibilities of different approaches for direct control and self-regulation, a careful analysis is required to make sure that integrated business processes are performed in a secure and smooth way. To this end, based on an existing agent based organization modeling approach OperA [1], we propose a framework that enables modeling and comparisons between different inter-organizational collaborative approaches.

2 A Context-aware Inter-organizational Collaboration Model

We first illustrate several concepts extended from OperA. A ***role*** is a set of objectives which indicate individual responsibility. To facilitate multi-level modeling from abstract to concrete, we define two kinds of roles: (1) atomic roles with relatively general objectives provide a macro-level understanding of what tasks will be carried out, and (2) composite roles with more details on how to accomplish the objectives through lower level organizations. An ***organization*** is a set of interdependent roles. There is only one top level organization marked as org_0 in each model and all the other organizations are derived from composite roles.

To provide actors with an evolutionary understanding of their responsibilities, a modeling process for organizational interactions is depicted in Fig. 1. First, a general specification is constructed to express the common objectives of inter-organizational collaborations in an abstract way. Then according to different contexts, the general specification is contextualized into different contextual specifications which describe

[1] Extended abstract of a paper originally published at EGOV 2011, Delft, The Netherlands.

the interactive relationships in a more detailed way and present the differences between different collaborative situations. Contextualization transforms some of the atomic roles to composite roles which contain more information on how to regulate the actors' behavior. Finally, the whole set of contextual specifications is transformed into different operational specifications which depict complete pictures of an inter-organizational collaboration model in different executable situations. That is, actors will match their status with the contexts in contextual specifications.

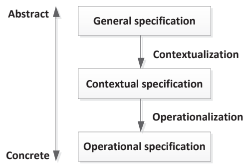
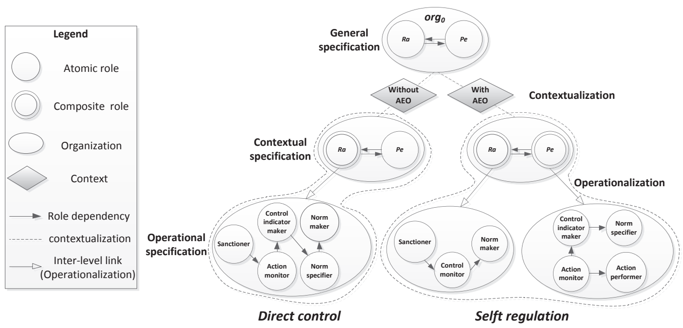


Fig. 2 shows an example to explain how *Regulatory authorities* (*Ra*) and *Private enterprise* (*Pe*) interact in two scenarios, direct control and self-regulation by a specific case of AEO [2] in international trade. At the top, *Ra* and *Pe* have the same interactive relationships at an abstract level. Contextualization generates concrete regulations according to different situations. In the direct control context (without AEO), *Ra* has to do most of the regulative tasks specifying by the five sub roles in the lower level organization. While in the self-regulation context (with AEO), some of the tasks shift to *Pe*. Each operational specification contains a complete description of organizational interactions associated with its context, which is an executable specification that can be seen as the assembling processes of different agents.



The proposed model describes inter-organizational interactions from abstract attitudes to concrete implementation, which supports users to understand their models during the procedure and reflect their design patterns even at the final operational stage. Moreover, it provides a potential solution to deal with the communication problems between multi-agents that jointly create shared norms.

References

1. Dignum, M.V.: A model for organizational interaction: based on agents, founded in logic. Doctoral Thesis. DSpace at Utrecht University (2004)
2. European Commission: AEO Guidelines, http://ec.europa.eu/taxation_customs/customs/

Applying Soft Systems Methodology in Enterprise Architecture Creation Workshops

A. (Agnes) Nakakawa[1], P. (Patrick) van Bommel[1], and H.A. (Erik) Proper[1,2]

[1]ICIS, Radboud University Nijmegen
P. O. BOX 9010 6500, GL Nijmegen, The Netherlands
[2]CITI, CRP Henri Tudor Luxembourg, Luxembourg
nakakawa@science.ru.nl, pvb@cs.ru.nl, e.proper@acm.org

The full paper of this abstract was published as "Nakakawa, A., Bommel, P. van, and Proper, H. A.: Applying Soft Systems Methodology in Enterprise Architecture Creation Workshops, in Proceedings of the 4th International Workshop on Enterprise Modelling and Information Systems Architectures (EMISA). M. Nttgens, O. Thomas, B. Weber (Eds.), Hamburg, Germany, September 22-23, 2011. LNI 190 GI 2011, ISBN 978-3-88579-284-0"
Keywords: Enterprise Architecture Creation, Soft Systems Methodology, Collaboration Engineering.

Extended Abstract

Lack of effective involvement of stakeholders is one of the common drawbacks of enterprise architecture development [1]. This paper, therefore, focuses on challenges associated with involving stakeholders in the enterprise architecture effort. These challenges are mainly caused by two issues, i.e.: (a) the success of collaborative sessions that involve enterprise architects and stakeholders mainly depends on the presence of a professional or skilled facilitator; (b) the lack of a clear, predictable, and repeatable way of managing tasks that require effective and active stakeholder involvement. Earlier attempts to overcome these issues involve using Collaboration Engineering to develop a process that enterprise architects can execute (by themselves) so as to manage tasks that require effective collaboration with stakeholders during enterprise architecture creation. Collaboration Engineering was chosen because it offers affordable facilitation to practitioners (in this case enterprise architects) of recurring high-value tasks (like enterprise architecture creation), by enabling the development of repeatable processes that practitioners can execute without hiring a professional facilitator [2,5].

According to [5], a collaborative process for a given task is designed using the following procedure: specifying the goal and deliverables of the process; defining the activities that participants must execute so as to achieve the goal; specifying the reasoning phases participants must undergo in order to achieve the goal; and describing detailed facilitation support for each activity. Facilitation support is specified by articulating: (a) the Group Support System (GSS) tools that should be used (or alternative techniques) during the collaborative sessions; (b) how the tools should be configured; and (c) the message prompts that should be followed [2]. This design approach was adapted when formulating the collaboration process for effectively involving stakeholders during enterprise architecture creation. This process is herein referred to as Collaborative Evaluation of Enterprise Architecture Design Alternatives (CEADA). The earlier version of CEADA was evaluated in a field study (of five organizations) where it was effective in supporting activities that required stakeholders to brainstorm, prioritize or rank or rate concerns and requirements for the architecture; and perform multi-criteria evaluation of possible enterprise architecture design alternatives. However, CEADA was still lacking adequate support for stirring vigorous and rigorous discussions when executing activities that required stakeholders and architects to reduce and organize aspects from brainstorming activities; and assess possible interrelationships and implications. This was reflected in the feedback from stakeholders who participated in the sessions supported by CEADA; the facilitator; and the observer of the sessions.

Since the main focus of this research is to offer effective stakeholder involvement in architecture creation, in this paper we address the above weakness by supplementing CEADA with techniques for enhancing the creation of a shared understanding and vision during execution of activities that involve organizing and discussing brainstormed aspects. We focus on adapting Soft Systems Methodology (SSM) because of its reputation for managing complex and ill-structured organizational problems through structuring rational thinking about them [3]. SSM techniques can be adapted to supplement the design of the collaboration process with support for triggering discussions and creating a shared understanding and vision among stakeholders. We also adapt the cause-effect analysis diagram (or Fishbone or Ishikawa) technique because of its support for thorough problem analysis [4]. Since SSM offers implicit facilitation support for collaborative workshops or discussion debates among problem owners and solver(s), Collaboration Engineering is further used in designing the facilitation script that shows how SSM and Ishikawa techniques can be used in enterprise architecture creation. Thus, in this paper CEADA is extended by a script that provides facilitation support for using SSM and Ishikawa diagram techniques to execute activities that require the use of clarify and organize patterns of reasoning.

References

1. Benard, S.A.: An Introduction to Enterprise Architecture - Linking Business and Technology. AuthorHouse, Indiana. (2005) ISBN 1420880500.
2. Briggs, R.O., Vreede, G.J. de, and Nunamaker, Jr. F.: Collaboration Engineering with ThinkLets to Pursue Sustained Success with Group Support Systems. Journal of Management Information Systems, 19:31–64, 2003.
3. Checkland, P.: Systems thinking, systems practice. John Wiley & Sons Ltd. Baffins Lane, Chichester (1998) ISBN-10: 0-471-98606-2
4. Ishikawa, K.: Guide to Quality Control. 2nd Edition. Asia Productivity Organization, Tokyo. (1986)
5. Vreede, G.J. de, and Briggs, R.O.: Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks. IEEE: HICSS (2005).

Change your lifestyle or your game is over

The design of a serious game for Diabetes

Harmen Nauta,

University of Twente, The Netherlands
harmennauta@gmail.com

Ton AM Spil,

University of Twente, The Netherlands
a.a.m.spil@utwente.nl

Abstract—Diabetes mellitus is one of the chronic diseases that leads to great concerns worldwide. This study uses a model of lifestyle change to enhance serious gaming in healthcare. Seven healthcare providers were interviewed in a pre-study, Six patients tested the prototype in a focus group and nine patients co-operated in an evaluation. The theoretical contributions show that it is difficult to design a single game that meets all users needs. One solution is to design a modular application were adaptations can be made in order to meet the patient specific needs, in the presence of the play, meaning and reality components, but also in functionality. Different functionality is needed for non-intenders (play), intenders (meaning) and actors (reality). Practical contributions of this study give the health care providers more insight in the daily activities and personal measurements of the patients and it improves the possibilities for self-management for the patients themselves.

a. *Which behavior methods are important to induce health behavior change and should be included in a serious game?*

The behavior methods ′education and skills', 'goal setting and action planning', 'self-monitoring′, 'reinforcements' and ′observational learning′, as presented in the model of lifestyle change and tested by diabetes type 2 patients, resulted to be important to induce a healthy lifestyle. Furthermore in the design of a serious game, the inclusion of these methods as potential game elements, appeared to be an effective approach.

b. *What distinction can be made in the population of diabetes type 2 patients regarding to the achievement of lifestyle change and what does this mean for a serious game?*

The division based on the motivation to change lifestyle, here the non-intenders, intenders and actors, was valuable in characterization of different groups in the diabetes type 2 population. From the findings it can be concluded that the major group of the diabetes population are intenders and consequently are the main target group for a serious game. However, for the non-intenders and actors serious gaming also has enough potential to support in the change or improvement of lifestyle.

c. *What is serious gaming in health care and which game components should be included in order to design a serious game?*

Serious gaming in healthcare needs to add a fun factor in the education or performance of health related task and is divided in health professional- and patient specific games. In the category of patient-specific games, a serious game to educate and coach a healthy lifestyle can include properties of the following type of games: education games, exergames and control games.

In order to design a serious game, the inclusion of the game components play, meaning and reality is important. However, in the actual design of a game, there are always tensions to what extent each of the components should be implemented. This is especially dependent on the target group, and what the actual meaning is of the game and from the game elements containing it.

> *Main Question - How can a serious game be designed for type 2 diabetes patients that both educates a healthy lifestyle and personally coaches during daily life? -*

Based on this study it can be concluded that in the design of a serious game to enhance lifestyle, the use of the model of lifestyle change was very valuable.

Regarding to the actual design of a serious game to educate and coach a healthy lifestyle, it can be concluded that the most important game elements are ´education and skills', 'goal setting and action planning', 'self-monitoring´, 'reinforcements' and ´observational learning. The extent to which the game components play, meaning and reality should be applied in the different game elements, will however differ per group of patients, defined in non-intenders, intenders and actors. While the inclusion of the play component in all game elements can be very valuable to the non-intenders to improve intrinsic motivation to change, for the intenders and actors especially in education it results to have an added value. In the other game elements the play component could generate an extra motivation, but this is individually dependent. In general this would mean it is difficult to design a single game that meets all users' needs. However, a solution is to design a modular application were adoptions can be made in order to meet the patient specific needs, in the presence of the game components, but also in functionality. A serious game as contemplated in this study can be an added value in the current provision of care, as it is one application that provides all tools in the support of reaching a healthy life style. It gives the health care providers more insight in the daily activities and personal measurements of the patients and it improves the possibilities for self-management. This results in more directed coaching and advises, also in between the quarterly controls. The intensity of use will differ for each group of patients. A patient can use it intensively in the first period after diagnosis, but when a healthy lifestyle is reached it will be used more like a reference. For the patients who are often monitoring their personal measurements, the application could be used even on daily basis.

In summary, the use of serious gaming to educate and coach a healthy lifestyle in daily life is a promising approach in the optimization of the diabetes care, which will be increasingly important in the coming years. The details of the application should however still be further investigated.

Keywords: Serious gaming; lifestyle change; diabetes type 2; E-health.

ACKNOWLEDGMENT

We thank Health @ Home BV for their support.