TOWARDS A FRAMEWORK FOR MULTIDIMENSIONAL QUALITATIVE ANALYSIS AND QUANTITATIVE ASSESSMENT OF BUSINESS MODELS

Paolo Pisciella, Darijus Strasunskas

Norwegian University of Science and Technology (NTNU), Dept. of Industrial Economics and Technology Management, NO-7491 Trondheim, Norway
{paolo.pisciella; darijus.strasunskas}@iot.ntnu.no

Abstract. An emergence of new technological aspects, as pervasiveness, ambience, service heterogeneity and dynamicity, opens new possibilities for business. However, information technology history testifies that the possession of the best technological solution is not necessarily enough to assure the business success of an enterprise. A very important aspect is the design and evaluation of appropriate business models. The task is not trivial since there is no data available. Therefore, this paper reports on a method for deriving quantitative valuation by mapping an existing qualitative business model with quantitative parameters. Qualitative attributes, those are typically available from market analysis and historical data of similar services, constitute the background to position a company under analysis. The contribution of the paper is a mapping of general business ontology to telecommunication domain with the purpose to evaluate cooperative advanced service provision. Furthermore, the mathematical model for deriving quantitative parameters from business model analysis is formalised in the paper.

Keywords: Qualitative analysis, quantitative assessment, business model, mathematical programming.

1 Introduction

With an emergence of new technological service aspects, as pervasiveness (ubiquity), ambience and context awareness, service heterogeneity and dynamicity, the importance of a suitable business model (BM) has increased. Pervasiveness of technology has brought similar trends into the business sphere in form of new complex service subscription models, service bundling, and collaboration in service provision, complex value configurations and business models [14]. Value creation in these new ubiquitous services is impossible without collaboration between participants. Typically, an actor (providing an enabler for a service) should choose between several service provision projects, each providing return on investment and generating cash flows (at a certain risk level). There is a dependency between service enablers, and each participant in such business network needs to balance own expectation with partners’ risk attitudes and return expectancies. These business and technical aspects challenge enterprise modelling in general, and especially business models that need to model aspects of collaboration, different value configurations and possible revenue share / revenue split.

Enterprise modelling seriously takes analysis of business and value creation [9], its processes and environment. There are approaches analysing goals of participants [11], networks of partners [6, 7], strategies [15]. However, most of these approaches are focused on qualitative evaluation. Some business model approaches (e.g., [6]) together with the related extensions encompass some sort of qualitative approach. However, they are often not sufficient to provide an accurate analysis of the company business logic, financial performance [3], which should be assessed in a quantitative manner in order to fine-tune the consequences of a given (set of) decision(s).

Actors engaged in a service provision should make a decision about the composition of their portfolio of services to which they are going to contribute. In [5] and [13] a method is proposed and implemented, it explains how actors can do this independently following the risk management framework of portfolio theory. However, the pricing and revenue sharing schemes induce the actors to contribute the right amount of provision capacity to participation in the service provision. The purpose of this paper is to provide a method that takes qualitative evaluation of business model a step further by identifying the quantitative outputs that are possible to achieve from qualitative analysis. This output then serves as an input to the method defined in [5, 13] and helps to balance partners’ cash-flow profiles and optimize Net Present Value (NPV).

Thus the objective of this paper is to analyse qualitative and quantitative methods suitable to evaluate business performance. Even though this analysis is referred to the problem of collaborative provision of a telecommunications data service, such an approach is quite general and easily suitable to other cases. The contribution of the paper is a mapping of existing qualitative business modelling approaches (i.e. business model ontology (BMO) [12]), to a formalised economical analysis. BMO is mapped to telecommunication domain in order to evaluate cooperative advanced service provision [13]. Then, the mathematical model for deriving quantitative parameters from qualitative business model analysis is presented.
The rest of the paper is structured as follows. First we provide a synopsis of related work in the areas of qualitative and quantitative evaluation of business models. Second, we present a model for optimization of net present value of actors participating in a collaborative service provision. Third, given synopsis of state-of-the-art and input information requirements for the proposed optimization model, we elaborate on mapping between a chosen qualitative BM ontology [12] and the proposed qualitative optimization model. Finally, we conclude the paper by discussing further work.

2 Related work

In order to understand rigorously the business of a company and set a proper business plan we need a business model to answer to a set of questions; some of the most important are as follows. What are the basic attributes of the service creating the customer value? (Value proposition); who is the customer? (Segments); how the Value Proposition is delivered to a given segment of customers? (Organisation and network of actors.)

Faber et al. [4] propose a framework for business modelling that encompasses four aspects of a given business model: a functional level (describing the service architecture), a strategic/organisational level (focusing on roles, actors and relations between actors as well as the physical and virtual flows between them), a financial level (analysing the revenue sources and costs flows for the actors involved), and a value proposition level (analysing the actual value created on the market, service portfolios and market segments of the actors). While, Leem et al. [10] propose a business model development methodology in ubiquitous computing environments that consists of 4 phases (i.e., business model planning, design, implementation and management), each of them having 14 activities composed of 26 detailed tasks. In business model design phase they propose to describe partners scenario defining business partners and describing partner’s activity scenario. However, that is not enough to model collaborative service provision and possible revenue split and share between participating parties. They propose a framework for the Business Model Feasibility Analysis (BMFA), consisting of two main parts: pre business model feasibility analysis and post business model feasibility analysis. Both parts have three aspects to analyse. The pre analysis is based on technological characteristics, requirements of users and enterprise strategies, while the post analysis assesses enterprise competencies, return on investment and risks.

Amongst the authors who tackle the development of business model considering some kind of evaluation metrics and indicators we find Afuah and Tucci [1]. They consider that a business model should include answers to a number of questions: What value to offer customers, which customers to provide the value to, how to price the value, who to charge for it, and what strategies to undertake in providing the value, how to provide that value, and how to sustain any advantage from providing the value. The business model approach they outline is value-centred and takes in account the creation of value through several actors.

Anyway, such kind of evaluation is mainly conducted comparing the quantitative indicators with the competitors’ ones, while it would be desirable to have a model describing which consequences a company can expect as output of given decisions.

A bit different approach to business models evaluation is outlined in [6, 7] and is part of their e3-value method. This methodology is based on a visual modelling of value creation processes. On the one hand it has the goal of improving communication and decision making related to e-business and on the other hand it aims at enhancing and sharpening the understanding of e-business operations and requirements through scenario analysis and quantification. Gordijn [6] proposes studying the economic feasibility of an e-business idea in quantitative terms by creating a profitability sheet and assessing the value of objects for all actors involved. This is possible because the method is highly actor-, network- and value-centred and focuses and the value exchanges among business model participants. The author admits [6] that this evaluation serves for building confidence in an e-business idea rather than calculating precise profit estimations, which would be unrealistic. Further, Gordijn introduces an additional confidence building step through the elaboration of “what-if” scenarios. This helps stakeholders understand the sensitivity of e-business models with respect to its parameters, such as future trends or customer behaviour.

Zoric & Strasunskas [16] discuss a framework for techno-business assessment. Four domain perspectives are used in the framework, as follows. User model describes services from the end-user perspective; business model conveys a conceptual framework for the business logic; system model provides complete details of the services from a system point-of-view; technical model exhibits technology and specifics of implementation.

The most work in the area of business modelling is focused on the qualitative analysis and evaluation of business capability, with an exception of the e3-value approach that generates profitability sheets at the end of analysis. However, it lacks periodisation in order to simulate a life-cycle of the service. In general, the e3-value approach is useful for initial analysis and visualization of the value activities. However, there is a need to simulate different possible situations and inter-dependence of the actors. The output provided by such approach is the quantitative evaluation of a given business model in terms of profits for actors involved [13].
The first step for analysis of business models is decomposition of business models by components. Osterwalder [12] provides a systematic and detailed business model ontology that allows to formulate a business model based on nine main components (Value Proposition, Target Customers, Distribution Channels, Relationship, Value Configuration, Capability, Partnership, Cost Structure, Revenue Model) grouped in four, so called, pillars (Product, Customer Interface, Infrastructure Management, Financial Aspects). Each of the components, eventually decomposed into sub-elements, is analysed with respect to a set of characteristics which describe, amongst other things, the relation between these elements and the related cardinality. Moreover each component is described by a set of attributes (direct or inherited from other elements or sub-elements), and for each attribute a qualitative evaluation is given. A broader description of the elements is provided in Table 1, though for more details the keen reader is referred to [12].

Table 1. Brief overview of BMO

<table>
<thead>
<tr>
<th>Business Model Ontology (pillars are described in grey rows)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product</strong> covers all aspects of what a firm offers its customers. This comprises not only the company’s bundles of products and services but the manner in which it differentiates itself from its competitors. <strong>Product</strong> is composed of the element <strong>Value proposition</strong> that can be decomposed into its elementary <strong>Offering</strong> (s).</td>
</tr>
<tr>
<td><strong>Value proposition</strong> represents value for one or several target customers and is based on one or several capabilities. It can be further decomposed into its sets of elementary offerings. A value proposition is characterized by its attributes Description, Reasoning, Value level and Price level and an optional Life cycle.</td>
</tr>
<tr>
<td><strong>Offering</strong> is a part of an overall Value proposition. It is characterized by its attributes Description, reasoning, life cycle, value level and price level.</td>
</tr>
<tr>
<td><strong>Customer interface</strong> covers all customer related aspects. This comprises the choice of a firm’s Target customers, the Channels through which it gets in touch with them and the kind of Relationships the company wants to establish with its customers. The Customer interface describes how and to whom it delivers its <strong>Value proposition</strong>, which is the firm’s bundle of products and services.</td>
</tr>
<tr>
<td><strong>Target customer</strong> defines the type of customers a company wants to address. The element can be further decomposed in criterion sub-elements.</td>
</tr>
<tr>
<td><strong>Criterion</strong> defines the characteristics of a target customer.</td>
</tr>
<tr>
<td>A distribution <strong>channel</strong> describes how a company delivers a value proposition to a target customer segment. Normally a company disposes of one or several direct or indirect channels that can be decomposed into their links.</td>
</tr>
<tr>
<td><strong>Link</strong> is a part of channel and describes a specific channel role. It may be a part of the value proposition and it may be related to another link.</td>
</tr>
<tr>
<td><strong>Relationship</strong> describes the relationship a company establishes with a target customer segment. Relationship is based on customer equity and can be decomposed into several relationship mechanisms.</td>
</tr>
<tr>
<td><strong>Mechanism</strong> describes a function it accomplishes between a company and its customer. It may also be a channel link or a part of the value proposition.</td>
</tr>
<tr>
<td><strong>Infrastructure management</strong> describes the value system configuration that is necessary to deliver the <strong>Value proposition</strong> and maintain customer interfaces. This comprises the <strong>Value configuration</strong> of the firm which can be decomposed in a set of basic Activities to create and deliver value and the relationship between them, the in-house <strong>Capabilities</strong> described by the sub-element <strong>Resources</strong> and those acquired through the firm’s <strong>Partnership network</strong>.</td>
</tr>
<tr>
<td><strong>Capability</strong> describes the ability to execute a repeatable pattern of actions. A company has to dispose of a number of capabilities to be able to offer its value proposition. Capabilities are based on a set of resources from the company or its partners.</td>
</tr>
<tr>
<td><strong>Resources</strong> are inputs into the value creation process. They are the source of the capabilities a company needs in order to provide its value proposition.</td>
</tr>
<tr>
<td><strong>Value configuration</strong> describes the arrangement of one or several activities in order to provide a value proposition.</td>
</tr>
<tr>
<td><strong>Activity</strong> is an action a company performs to do business and achieve its goals.</td>
</tr>
<tr>
<td><strong>Partnership</strong> is a voluntarily initiated cooperative agreement formed between two or more independent companies in order to carry out a project or specific activity jointly by coordinating the necessary capabilities, resources and activities. They are based on a range of <strong>agreement</strong> sub-elements.</td>
</tr>
<tr>
<td><strong>Agreement</strong> specifies the function and the terms and conditions of a partnership with an actor.</td>
</tr>
<tr>
<td><strong>Financial aspects</strong> is the last block of BMO framework: It is transversal because all other pillars influence it. This block is the outcome of the rest of the business model's configuration. Financial aspects are composed of the company’s <strong>Revenue model</strong> and its <strong>Cost structure</strong>. Together they determine the firm’s profit- or loss-making logic and therefore its ability to survive in competition.</td>
</tr>
<tr>
<td><strong>Revenue model</strong> describes the way company makes money.</td>
</tr>
<tr>
<td><strong>Revenue stream and pricing</strong> describes an incoming money stream from the value offered by the company. Furthermore, it defines what mechanism is used to determine the price of this value offered.</td>
</tr>
<tr>
<td><strong>Cost structure</strong> measures all monetary costs incurred by the company.</td>
</tr>
<tr>
<td><strong>Account</strong> is a registry of pecuniary transactions (expenditures) of a certain category.</td>
</tr>
</tbody>
</table>

Attributes are then used to perform comparisons with the competitors and to position the company in the market. Such ontology allows analysing business models in order to have a clear view of the positioning of
the company in the market. For what concerns the financial aspects, the approach still leads to describe the revenue sources and the cost structure in a qualitative point of view.

3 A Model for Cooperative Service Provision Performance Optimization

Mobile service provision in the telecommunication sector may take different forms with respect to the roles taken by various participating actors [2, 14]. However, there are two main types of actors in the domain – Component providers (enablers) and a Platform operator. For each actor we describe the environment by the means of a set of elements that are linked to denote dependence between them, i.e. one element influencing another. We differentiate between three types of elements: input, actor decision and output (see legend of Fig. 1). Component providers are trying to position in a reasonable way the variables, the decisions and the consequences, bundling all of this in an overall picture of the environment (Fig. 1). Note that everything is related to a single actor (the component provider) even though we may speak of multiple services. Similar description is provided for the platform operator\(^1\) in Fig. 2.

In both figures the \textit{NPV optimization} is considered as the final problem of the Component provider. NPV optimization is influenced by the values of Outflows and Inflows (i.e. incomes and outcomes). There is a set of inputs that provides a boundary for the actors, as follows. \textit{Demand} is considered external, which means that the level of demand for a platform service is not directly influenced by Component provider’s decisions. However, it influences amount of the component provided for that service; component capacity since capacity needs to be adapted to the changes of the demand over time; risk due to its intrinsic stochasticity.

Furthermore, \textit{price of the service} together with the \textit{sharing scheme} influence (given the level of the demand) the \textit{amount of component provided} by the component provider for a given service. A higher level of return (in terms of price per sharing scheme) will make the component provider to supply more components (allocate more capacity). The level of \textit{Risk} (if we use standard deviation as a risk measure it changes linearly with the sharing scheme percentage for the given service) is also influenced by the price and sharing scheme. \textit{Maximum risk} provides a constraint on the maximum level of risk that a component provider is willing to take.

While, actor decisions in this perspective are as follows. \textit{Component capacity} constrains the \textit{amount of component provided}; the \textit{amount of outflows} as capacity building generates capital expenditures over time. At the same time \textit{amount of component provided} has costs associated with the level of provided services and, therefore, constrains the level of the outflows; \textit{the inflows} together with the price and sharing scheme per service. Furthermore, the risk profile of the portfolio of services changes accordingly to the level of investments in each service.

![Figure 1. Interdependencies in the Component provider environment](image)

However, in the Platform operator environment this problem deals with inputs and outputs provided by multiple actors. There the \textit{amount of component provided} and the \textit{component capacity} are seen as outputs in the model, and they are maximized in the Component provider modelling stage. Here, inputs are as follows.

\(^1\) Although we remark that the Platform operator is a component provider as well. This means that he has to solve the Component provider problem as well.
Minimum price for the component influences the level of share of the prices that the various component providers are willing to accept given by the product of price of the service and sharing scheme; maximum risk constrains the level of risk that the component providers are willing to accept; minimum component level constrains the level of component that each Component provider has to provide in order for a given service to be feasible.

Consequently, actor decisions to be made are as follows. Price of the service has exclusively effect on the demand of the service, determining the amount of service sold; the price of the service together with the sharing scheme influences (given the demand) the amount of component provided. We can expect that a higher level of return (in terms of price per sharing scheme) will make the component provider to provide more components.

4 Our Proposal to Mapping of BMO and Quantitative Analysis Model

Business models, due to the way they structure a business activity, can help estimating the parameters involved in the mathematical description of a given problem. In particular, whenever a business model allows for decomposing the business structure of a firm in blocks and performing a standardized set of evaluations for each block considered, it is possible to use the set of evaluations to establish a link between such evaluations and the parameters of a quantitative model. Osterwalder’s BMO is probably not the unique framework that can be used to achieve this goal, but since it allows for decomposition of the business of a firm, followed by the evaluation of each building block, it is well suited for the analysis. In next sub-section we elaborate on the mapping between BMO [12] (briefly introduced in Section 2) and the optimization models described above (Section 3). Then we formalise a quantitative model to evaluate business models.

4.1 Mapping of BMO to analytical model of service provision

A way to link BM analysed by the means of BMO with a formal mathematical model could be by assigning to each main set of equations in the model (diffusion, demand, costs) a pillar or a subset of elements composing the Osterwalder based business model. As mentioned a link can be established between the diffusion curve parameters and the “Product” pillar in BMO, which in this case would be directly expressed by the Value Proposition element. A second link can be established between the “Customer Interface” pillar, which is mostly related to the customer relationship and the communication with the customer, with the parameters in the demand function. A last link could connect the “Infrastructure Management” pillar, which is based on the
type of activity, the capability of the company in relation to the activity and the bargaining power analysed in the partnership network element, with the cost modelling.

We will perform this analysis in two steps: the first step will be focused on relating an existing ontology’s elements [12] to a set of quantitative oriented elements, and the second step will involve the study of the connections between the previously introduced elements distinguishing two types of environments, namely the Component provider and the Platform operator environments.

The basis is relating the qualitative BMO to the models where each element of the model is related to one or more elements of BMO. In this way, the qualitative elements influence the parameters of a mathematical formalization of our models by the means of the attributes they are characterized by in BMO. Consequently, we introduce the elements of the model distinguishing them in demand side and supply side elements. Each element is composed by sub-elements which can be modelled separately and can be related to different BMO elements by the means of different attributes of such latter elements.

An element of our model can as well be a sub-element of other elements of the model. In this case the element containing the other element will be influenced by the same BMO elements which influence the sub-element. The level of demand is the only element we consider from the demand side perspective. It is characterized by four sub-elements: Market share, Speed of adoption, Decline time, Usage.

OFFERING\(^2\) describes the level of Market share, Decline time and Usage depending on the way the service fulfils the needs of the customer and its positioning w.r.t. the competitors, particularly in relation to the decline time and the usage, by the means of the attributes (price level, and value level). LINK influences the level of DEMAND due to the same attributes (the LINK element can be also an offering in BMO), but we assume that it works mainly in the introduction phase of the service life-cycle. So the Speed of adoption will be mainly influenced, together with Market share and Usage. MECHANISM element in BMO describes the way in which the relationship with the customer is made up. It is crucial for maintaining customers’ loyalty over time. We assume than that this element influences mainly the Decline time and the Market share level. VALUE CONFIGURATION is modelled considering mainly the form of control in the structure of the production system. We assume that a high level control is focused in cost efficiency (value chain) while a low level control has the aim of technology and process model improvement (value shop and network). Under the DEMAND viewpoint it affects the Market share, given the price of the service, when the output of the production process is technologically better than the competitors. AGREEMENT element defines the terms and conditions for a partnership with an actor. It can affect the Market share smoothing the competition and improving the knowledge within the process. Finally, REVENUE STREAM AND PRICING affects the Market share accordingly to the pricing scheme used.

Supply side perspective is modelled by quantity, pricing and randomness (i.e. risk). Therefore, the component capacity is chosen before the actual service is issued, and it depends by the sub-elements Demand, Level of component outcome, Overall capacity per tool, Cost per tool. Demand here is influenced by the BMO elements: OFFERING, LINK, MECHANISM, VALUE CONFIGURATION, AGREEMENT and REVENUE STREAM AND PRICING; such elements influence the COMPONENT CAPACITY level as well. Moreover, RESOURCE affects the Level of component outcome depending on the efficiency level of such resources. AGREEMENT has its effect on the Level of component outcome since it can lead to a more efficient usage of the capacity.

While the amount of component provided by a given component provider to the service bundling is chosen at each service issue and depends on the sub-elements Demand, Component capacity, Risk, Costs per component. As previously, the sub-elements Demand, Component capacity and Risk are also elements of our model, so the set of BMO elements influencing these elements, influence the amount of component provided as well. VALUE CONFIGURATION, accordingly to the structure of the company and the level of control affects the level of costs, in this case the Cost per component is affected considering the level of control embedded in the system.

Minimum component level is a technical constraint and describes the minimum level of a given component in order to provide one instance of service. We assume such a parameter as given with the sub-element Level.

Pricing and revenues are determined by the price of the service that is a decision taken by the company and is the result of a set of considerations on the sub-elements Demand, Component capacity, Minimum component level, Minimum price which should optimize the profitability for a given level of the other variables. Its other function is to act as a matching variable between level of demand and level of component provided by the component providers in order to make feasible the desired level of service that the company is willing to provide. OFFERING affects Demand and Component capacity, just like LINK and MECHANISM. RESOURCE affects Minimum component level and Component capacity. VALUE CONFIGURATION affects Demand, Component capacity and Minimum component level. AGREEMENT influences Demand and Component

\(^2\) We use capital letter to denote constructs from BMO.
capacity. REVENUE STREAM AND PRICING influences the level of Demand, Minimum price and Component capacity.

We assume that the minimum price for a component is a given parameter (linked to some break-even analysis or to some comparable services in the market) and it is characterized by its sub-element Level linked to the BMO element REVENUE STREAM AND PRICING. Furthermore, the Sharing scheme component has approximately the same function as the choice of the price but is more focusing towards the collaborative provision. The aim is to ensure that every component provider gives enough component (a quantity larger than the minimum component level) in order to make the desired level of services provision feasible. The sub-elements of this element are the same as the ones characterizing the element price: Demand, Component capacity, Minimum component level, Minimum price. Therefore, the element is linked to the same BMO elements as price of the service is linked to.

Finally, randomness is modelled by risk. Risk is considered as uncertainty of future outcomes and it is modelled considering, for each parameter estimated, the connection between the variance of such a parameter estimated in different time frames around the mean value of each parameter estimated. Then a series of simulations will lead to obtain the estimation of the risk on the revenue level. Level of risk is the only sub-element composing this element and, being this element recovered by simulations involving every parameter we have that each element of the BMO influence, even though indirectly, this element of our model. Consequently, maximum risk describes the maximum level of risk that a component provider is willing to accept. It is characterized by the sub-element Level and is related to the BMO element REVENUE STREAM AND PRICING. Table 2 summarize the mapping.

Table 2. Linking of BMO elements to our model

<table>
<thead>
<tr>
<th>Our model</th>
<th>BMO elements</th>
<th>Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demand</td>
<td>Offering</td>
<td>- Market share [REASONING, VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Decline time [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Usage [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td>Criterion</td>
<td></td>
<td>We assume such influences of BMO for a given market segment</td>
</tr>
<tr>
<td>Link</td>
<td></td>
<td>- Market share [REASONING, VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Speed of adoption [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Usage [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td>Mechanism</td>
<td></td>
<td>- Cannibalization time [FUNCTION, VALUE LEVEL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Market share [REASONING, CUSTOMER EQUITY]</td>
</tr>
<tr>
<td>Value configuration</td>
<td></td>
<td>- Market share [CONTROL]</td>
</tr>
<tr>
<td>Agreement</td>
<td></td>
<td>- Market share [DEGREE OF COMPETITION, SOSTITUTABILITY]</td>
</tr>
<tr>
<td>Revenue stream and</td>
<td></td>
<td>- market share [PRICING METHOD]</td>
</tr>
<tr>
<td>pricing</td>
<td></td>
<td>- speed of adoption [PRICING METHOD]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- cannibalization time [PRICING METHOD]</td>
</tr>
<tr>
<td>Component capacity</td>
<td>Offering</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td>Link</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td>Mechanism</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td>- level of component outcome [RESOURCE TYPE]</td>
</tr>
<tr>
<td></td>
<td>Value configuration</td>
<td>- Cost per tool [CONTROL]</td>
</tr>
<tr>
<td></td>
<td>Agreement</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- overall capacity per tool [REASONING, DEGREE OF INTEGRATION]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- level of component outcome [REASONING, DEGREE OF INTEGRATION]</td>
</tr>
<tr>
<td></td>
<td>Revenue stream and pricing</td>
<td>- Price of the service * (modelled in PRICE OF THE SERVICE)</td>
</tr>
<tr>
<td></td>
<td>Account</td>
<td>- Cost per tool [SUM, PERCENTAGE]</td>
</tr>
<tr>
<td>Amount of component</td>
<td>Offering</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td>provided</td>
<td></td>
<td>- Component Capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Risk* (modelled in RISK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Link</td>
<td></td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component Capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Risk* (modelled in RISK)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Mechanism</td>
<td></td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component Capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Resource</td>
<td></td>
<td>- Costs per Component [RESOURCE TYPE]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Value configuration</td>
<td></td>
<td>- Costs per Component [CONTROL]</td>
</tr>
<tr>
<td>Our model</td>
<td>BMO elements</td>
<td>Mapping</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Agreement</td>
<td>- Demand* (modelled in DEMAND)</td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Revenue stream</td>
<td>- Demand* (modelled in DEMAND)</td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Account</td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Minimum component</td>
<td>Resource</td>
<td>- Level* (not modelled with attributes)</td>
</tr>
<tr>
<td>Price of service</td>
<td>Offering</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Link</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Mechanism</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td>- Minimum Component Level* (modelled in MINIMUM COMPONENT LEVEL)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Value configuration</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Agreement</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Revenue stream and pricing</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum price (modelled in MINIMUM PRICE FOR THE COMPONENT) - Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Account</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td>Sharing scheme</td>
<td>Resource</td>
<td>- Level* (not modelled with attributes)</td>
</tr>
<tr>
<td></td>
<td>Offering</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Component capacity* (modelled in COMPONENT CAPACITY)</td>
</tr>
<tr>
<td></td>
<td>Link</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td>Resource</td>
<td>- Minimum Component Level* (modelled in MINIMUM COMPONENT LEVEL)</td>
</tr>
<tr>
<td></td>
<td>Value configuration</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Cost per tool [CONTROL]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Costs per Component [CONTROL]</td>
</tr>
<tr>
<td></td>
<td>Agreement</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td></td>
<td>Revenue stream and pricing</td>
<td>- Demand* (modelled in DEMAND)</td>
</tr>
<tr>
<td>Risk</td>
<td>Offering</td>
<td>- Level of risk [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td>Link</td>
<td>- Level of risk [VALUE LEVEL, PRICE LEVEL]</td>
</tr>
<tr>
<td></td>
<td>Value configuration</td>
<td>- Level of risk [CONTROL]</td>
</tr>
<tr>
<td></td>
<td>Agreement</td>
<td>- Level of risk [REASONING, STRATEGIC IMPORTANCE, DEGREE OF INTEGRATION, DEGREE OF COMPETITION, SOSTITUTABILITY]</td>
</tr>
</tbody>
</table>

Note that the elements of the model are linked to the elements of BMO by the means of a set of attributes. It is possible to classify such attributes in two categories concerning the role they play in relating the qualitative business model to a quantitative approach as follows: attributes that cannot be ordered, but are useful for positioning the service in relation to the other services (positioning attributes), and attributes for which an ordering can be established and can be used in the quantitative evaluation phase (evaluation attributes). The former ones are used to position the service the company wants to issue in relation with comparable projects and to establish a relation between evaluations of specific attributes. Only higher level elements are considered here, and for each element the positioning attributes are used to assign a cluster of firms sharing the same qualitative evaluations regarding such attributes. The latter ones are used to perform an estimation of the parameters of a formal mathematical model for the element of the model they are related via the linking with the set of BMO attributes, as shown in Table 2.

Furthermore, we distinguish between technical and business attributes: the first ones are focused on the level of innovation, performance in satisfying needs and limitability of the service and the second ones are more focused on market peculiarities and economic-financial choices (level of competition, level of control on the firm activities, pricing scheme etc.). This classification is shown in Table 3.
Table 3. Techno-Business and analysis oriented Classification of BMO attributes

<table>
<thead>
<tr>
<th>Positioning attributes</th>
<th>Technical attributes</th>
<th>Business Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Reasoning</td>
<td>Pricing method</td>
</tr>
<tr>
<td></td>
<td>Customer equity</td>
<td>Stream type</td>
</tr>
<tr>
<td>Valuation attributes</td>
<td>Value level</td>
<td>Price level</td>
</tr>
<tr>
<td></td>
<td>Strategic importance</td>
<td>Level of control</td>
</tr>
<tr>
<td></td>
<td>Substitutability</td>
<td>Degree of competition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Percentage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Sum</td>
</tr>
</tbody>
</table>

Each element of the model is quantitative in the sense that it is transformable into a number. Such a number is generally a function of a set of inputs defined in the by the element. The considered function can be a simple analytic function (typically when the element does not describe a decision) or even a mathematical programming problem (when the element describes the output of a decision, which typically requires an optimization approach in order to maximize or minimize a given objective function with the decision variables constrained by a certain set).

4.2 A model for quantitative analysis

Here we introduce a formalization of the connection between the qualitative BMO and a mathematical model following the framework provided by the elements of our model (Figures 1 and 2). Therefore, a specific element in the model is defined by a set of parameters that are estimated via the existing data about input chosen and related output. In the first stage of the estimation analysis we use the positioning attributes in order to perform a clustering of the services. We can assess new services considering a set of comparable services. A cluster of the comparable services is made by computing distances between attribute values of corresponding analysis elements. Assuming that the cluster has \(N+1\) services (\(N\) services for which historical data is available and the service we want to evaluate), we can use these services to collect measurements of \(M\) (BMO evaluation attributes), both technical and business related, relevant for the given element \(e\) of the model. Consequently, we are placing them in the following techno-business attributes matrix, equipped with a unit vector as the first column:

\[
\Omega = \begin{pmatrix}
1 & \omega_{1,1} & \cdots & \omega_{1,N} \\
1 & \omega_{2,1} & \cdots & \omega_{2,N} \\
\vdots & \vdots & \ddots & \vdots \\
1 & \omega_{M,1} & \cdots & \omega_{M,N}
\end{pmatrix}
\]

Let the generic parameter we want to model be \(p_j\) in order to describe a given element \(e\) of the model. We assume each parameter \(p_j\) of the model, with the exception of the parameters linked to dynamic considerations (as the ones modelling the diffusion of the service), can be estimated at different time (i.e. different parameters \(p_j\) in different timeframes \(t_1,t_2,\ldots,t_T\)), obtaining then a set of historical data for each parameter. Further, we can model each parameter used in the model for the element \(e\) and related to the service \(j\) with Eq. 1.

\[
p_j = a_j + b_j p_i + \epsilon_j
\]  

(1)

Where \(p_i\) denotes an index for the parameter obtained by a “proper” convex linear combination of the same parameter considered for the other services. We have that

\[
b_j = \frac{\text{cov}(p_j; p_i)}{\sigma_i^2}
\]

and

\[
a_j = \mu_j - b_j \mu_i
\]

where \(\mu_j\) is an average value of the parameter \(p_j\). Assuming \(\epsilon_j \sim N(0; \sigma_i^2)\) and \(\text{cov}(\epsilon_i; \epsilon_j) = 0\), if \(i \neq j\).

Let us further introduce another matrix describing what BMO attributes are active when working with a given element of our model. We do this using a matrix containing as columns the basis vectors \(e_m\), where \(m\)

\[\text{Note, that a basis vector } e_m \text{ is a vector containing all zeros except for the element in position } m, \text{ which is set to the unit value.}\]
denotes the index of the BMO evaluation attribute used for the estimation of the parameters linked to the model element \( e \)

\[
\mathbf{H} = \left[ e_i | e_m \right] \ \forall m \in I \left(A_i(e)\right)
\]

where \( I(A_i(e)) \) is the set of the indices of the BMO evaluation attributes affecting the parameter \( p_j \) in the evaluation of the model element \( e \). The matrix used for the estimation of the parameter \( p_j \) in the evaluation of the model element \( e \) is the reduced techno-business matrix, given by

\[
\Omega = \Omega \mathbf{H}
\]

Now, we use the matrix \( \Omega \) in order to estimate three sets of coefficients: the first set will be used to obtain the \( a_j \) coefficient related to a given service \( j \); the second set will provide the \( b_j \) coefficient; and the third set will be used to estimate the standard deviation of the random noise related to the particular parameter for that service.

Denoting respectively with \( \mathbf{a}, \mathbf{b} \) and \( \sigma \), the vector of the \( a_j, b_j \) coefficients for each service \( j \) and the standard deviation of the random noise \( \sigma_{\epsilon_j} \), we can calibrate the model in order to obtain coefficients to predict parameters accordingly to Eq. 1 using a multiple linear regression approach, see Eq. 2

\[
\beta_v = \left( \Omega^T \Omega \right)^{-1} \Omega^T \mathbf{v}
\]

for \( \mathbf{v} = \mathbf{a}, \mathbf{b}, \sigma \in \mathbb{R}^N \).

Denoting the vector of BMO evaluation attributes for the service \( j \) as

\[
\omega_j = \begin{bmatrix} a_j \mid b_j \mid \sigma_{\epsilon_j} \end{bmatrix}
\]

We can now obtain the statistics for a given service using the following formulae:

\[
\mu_j = a_j + b_j \mu_t
\]

\[
\sigma_j = \sqrt{b_j^2 \sigma_t^2 + \sigma_{\epsilon_j}^2}
\]

and the linear correlation coefficient given in Eq. 6

\[
\rho_{\epsilon} = \frac{b_j \sigma_{\epsilon}}{\sigma_j}
\]

where \( \rho_{\epsilon} = \frac{b_j \sigma_{\epsilon}}{\sigma_j} \).

At this point we can turn our attention to how to select a “proper” convex linear combination of service parameters \( p_j \) to build an index \( p_I \). An idea is to take the set of parameters that minimizes the variance of the estimation error \( \sigma_{\epsilon_j} \) for the parameter \( p_j \) in order to have the standard deviation explained as much as possible by the product of \( |b_j| \sigma_j \). The problem is given by finding the vector of weights \( \mathbf{q} \) to minimize the estimation error \( \sigma_{\epsilon} \) for the parameter \( p_j \), or formally:

\[
\min_{\mathbf{q}} \sum_j \sigma_{\epsilon_j}^2
\]

From Eq. 5 we have that

\[
\sigma_{\epsilon_j}^2 = \sigma_j^2 - b_j^2 \sigma_t^2
\]

and being \( \sigma_j^2 \) a fixed value we have that

\[
\min_{\mathbf{q}} \sum_j \sigma_{\epsilon_j}^2
\]

is equivalent to the problem

\[
\max_{\mathbf{q}} \sum_j b_j^2 \sigma_t^2.
\]

The whole problem is expressed as the following mathematical programming problem

\[
\min_{\mathbf{q}} \frac{\left( \sum_j q_j \sigma_{\epsilon_j} \rho_{\epsilon} \right)^2}{\sum_j \sum_i q_j q_i \sigma_{\epsilon_j} \rho_{\epsilon_j}}
\]

sub constraints \( \sum_i q_i = 1 \)

\( q_i \geq 0 \)

- 70 -
resulting in a minimum error method for index search.

In summary, Eq. 1 defines the generic parameter of the model related to a given element of the model as a random variable that is subject to variations over time. Thus it can be described by two parameters: mean and standard deviation. Parameters $a_j$, $b_j$ and $\sigma_j$ for the firms with quantitative market data available can be estimated using historical data. Then we use the reduced techno-business matrix (i.e. the matrix of techno-business evaluation BMO attributes multiplied by matrix $H$ that is used to get rid of the columns describing the evaluation attributes not used to model for a specific element in the model) to perform a multiple linear regression analysis to tune the BMO evaluation attributes in order to estimate the coefficients $a_j$, $b_j$ and $\sigma_j$ to be used in Eq. 1 to assess the relevant parameter $p_j$.

5 Conclusions and future work

The paper reports on a method for deriving quantitative valuation of enterprise operational performance by bridging qualitative business model analysis with quantitative parameters. This method demonstrates how enterprise models could be used to derive quantitative measures of business performance. Here we have focused on mapping business model ontology [12]. However, the proposed linkage between a qualitative business analysis model and inputs needed for the quantitative assessment of business model could feasibly be adapted by other enterprise modelling approaches in order to compute the numerical value of enterprise operations.

Nevertheless, we need to empirically validate our model. Therefore we are now collecting the data for the case study. As well the model should be extended to incorporate simulations that would allow balancing expectation of a component provider with partners’ risk attitudes and return expectancies, where the goal is to optimise net present value of each participant.

References

THE APPROACH OF TRANSFORMATION BETWEEN BUSINESS PROCESS DIMENSIONS IN BPMN MODELING TOOL

Ludmila Penicina

Riga Technical University, Department of Systems Theory and Design, 1 Kalku, Riga, LV-1658 Latvia, ludmila.penicina@rtu.lv

Abstract. Most popular business process modeling tools that support BPMN standard provide functionality of creating one or two dimensional process models, however, without possibility to transform created BPMN model to other dimensions. The paper describes the approach of switching between business process dimensions that could be implemented in BPMN modeling tool as advanced feature. The functionality of switching between process dimensions may contribute to the business process modeling and analysis from different perspectives in one environment. To make the transformation between business process model dimensions possible it is required to define additional attributes of the BPMN model graphical elements. The aim of the paper is to describe how BPMN model elements need to be parameterized for switching between dimensions in any BPMN modeling tool.

Keywords: business process modeling, modeling dimensions, BPMN, modeling tool.

1 Introduction

A business process is a set of coordinated activities that are performed either by people or by tools with an objective to realize a certain business result [1]. To define how the process works it is crucial to develop the process model, which will define the existing process flow in detail. Knowing and understanding the details of business processes is important, because this gives the opportunity to identify the bottlenecks and optimize business processes. To reveal all the details of the business process it is necessary to analyze the process from different point of views, within different contexts, i.e., from different dimensions/perspectives. It means that the process that is represented, e.g., from performer’s dimension is hiding in its structure the details about the process from the time perspective. In most of the BPMN tools the only way to obtain business process model constructed from a particular perspective in other dimension is to redraw the process diagram. Apparently this solution is time consuming, not effective and may create errors in the process semantics.

The paper presents the idea of automatic business process model transformation between process dimensions within a BPMN modeling tool. The BPMN modeling tool is any tool that support BPMN standard, however in for obtaining model in several dimensions it is necessary that the tool allows to define additional attributes to the graphical elements as the idea of transformation between dimension is based on the refinement of the BPMN model by parameterizing its elements with additional information. BPMN is a chosen standard because it uses the concept known as “swimlanes” to help partition and organize activities. The idea of presenting business process in a new dimension is to organize model elements in the different way using swimlanes in order to reveal important details typical to the new dimension. The other reason is that BPMN quickly established itself as the standard notation for modeling executable business processes [1].

The paper is structured as follows. The overview of tools supporting multidimensional business process modeling is presented in Section 2. The approach of the transformation between dimensions is analyzed in Section 3. The requirements for the modeling tool with multidimensional process modeling functionality are presented in Section 4. The experimental example of transformation between process dimensions is presented in Section 5. Conclusions and future work is described in Section 6.

2 Related works

Multidimensional modeling capabilities of current business process modeling tools are quite limited in a sense that in most cases changing modeling dimension or perspective means creating new business process models. Only few tools allow limited automatic transformations for multidimensional modeling purposes [2]. One of such tools is JOpera [6] that supports visualization of control and data flow views, however these representations are independent and cannot be transformed one to another. In JOpera tool [6] the control and data flow graphs of a business process are displayed and edited separately [2]. Multidimensional modeling capabilities of BPMN tools are even more limited and in most cases they support multidimensional modeling by allowing two lanes to overlap (e.g. horizontal and vertical lane) – in that way creating two dimensions – such tools are MagicDraw [7] and ProcessModeler 5 [8].

Several business process analysis tools, such as Oracle BPA [9] provide capabilities for simulating processes [1]. Business process simulation starts by adding special parameters to process activities and events –
this refinement would allow to carry out simulation [1]. This idea of refinement business process model with special attributes to enable transformation is used in this article.

3 The approach of the transformation

3.1 Determining dimensions

In business process modeling phase it is essential for process model to reveal the following basic information about each of the activity performed in the business process [1]:
1. Roles responsible for carrying out each activity in the process
2. Start time/end time of the process and processing time of each activity in the process
3. Documents exchanged within the process (inputs and outputs of each activity),
4. Business rules that control the workflow

In this paper each of above mentioned slots of information is regarded as business process modeling dimension. According to each of these dimensions a process model can be created and then transformed to another dimension without losing process semantics, using BPMN syntax and revealing information about the process in new context. Table 1 describes the business process dimensions in more detail.

Table 1. Descriptions of dimensions

<table>
<thead>
<tr>
<th>Dimension name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performer dimension</td>
<td>Specifies which resource is responsible for carrying out each activity in the process. The performer dimension may involve human actors, devices and systems.</td>
</tr>
<tr>
<td>Time dimension</td>
<td>Defines the execution order of activities in the process according to time granularity (months, weeks, days, hours and so on) chosen by modeler.</td>
</tr>
<tr>
<td>Business rule dimension</td>
<td>Defines which activities are controlled by business rules and where in the process key business decisions are made.</td>
</tr>
<tr>
<td>Information dimension</td>
<td>Defines activities which use for input artifacts produced in the business process and activities that create as their output business artifacts.</td>
</tr>
</tbody>
</table>

3.2 Correlation between BPMN elements and dimensions

BPMN is a visual language used for business process modeling, and uses a set of graphical elements. BPMN provides four basic categories of elements [3]:
1. Flow objects – events, activities, gateways
2. Connecting objects – sequence flow, message flow, association
3. Swimlanes – pools, lanes

In this paper BPMN is selected as the notation for representing business process model in different dimensions based on the following statement in the BPMN specification [3] – “within the basic categories of elements, additional variation and information can be added to support the requirements for complexity without dramatically changing the basic look and feel of the diagram”. From this statement it can be concluded that the idea of BPMN model refinement by adding special parameters to graphical elements to allow transformation between dimensions is not in conflict with BPMN specification. The second reason for proposing BPMN as official multidimensional modeling standard in this paper is the statement in BPMN specification [3] that describes a pool as the container for the sequence flow between activities, so a pool or a lane can be considered as a container for the flow of activities according to a particular dimension.

Table 2 describes how business process dimensions correlate with BPMN basic elements and what additional attributes must be defined to these elements in order to make transformation to each possible dimension.
Table 2. BPMN correlation with dimensions

<table>
<thead>
<tr>
<th>Dimension name</th>
<th>BPMN element</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performer dimension</td>
<td>Pools, lanes</td>
<td>Define the names of the pool and lanes.</td>
</tr>
<tr>
<td>Time dimension</td>
<td>Start event</td>
<td>Define process start time by adding time attribute to the start event.</td>
</tr>
<tr>
<td></td>
<td>End event</td>
<td>Define process end time by adding time attribute to the end event.</td>
</tr>
<tr>
<td></td>
<td>Intermediate event</td>
<td>Define the wait time before a particular activity can be initialized by adding time attribute to the intermediate event.</td>
</tr>
<tr>
<td></td>
<td>Task</td>
<td>Define the processing time of each activity by adding time attribute to the activity element. This time indicates the actual time spent for the activity.</td>
</tr>
<tr>
<td>Business rule dimension</td>
<td>Gateways</td>
<td>Define the condition for executing process flow by adding name attribute to the gateway element.</td>
</tr>
<tr>
<td>Information dimension</td>
<td>Data object</td>
<td>Define the name of the artifact by adding the name attribute to the artifact element.</td>
</tr>
</tbody>
</table>

4 The requirements for the modeling tool

There are two basic preconditions for business process model transformation to another dimension, namely, the original business process model has to be constructed in the tool (1) with implemented dimension transformation feature and (2) compatible with BPMN version 1.1 standard. In this section the requirements for BPMN modeling tool are described to support dimension transformation functionality.

4.1 Transformation to time dimension

To transform business process model from any dimension to time dimension it is required to define:

1. Start time of the process using BPMN start event element,
2. End time of the process using BPMN end event element,
3. Wait time before a particular activity can be started using BPMN intermediate event,
4. Processing time of each activity using BPMN activity task.

The time should be defined in a format dddd:hh:mm:ss (e.g. 0002:02:23:54, which means that activity is performed 2 days 2 hours 23 minutes and 54 seconds) [9]. The start and/or end time should be defined as a date and time.

BPMN 1.1 offers different types of activities, so processing time of the process activity must be defined according to its type:

1. if the activity is a looping task – then the processing time of one activity instance must be defined and then multiplied by loop counter,
2. if the activity is a multi-instance – then the processing time of one activity instance and the number of instances to be performed must be defined.

It should be possible for the user to define the granularity of the time dimension before transformation – in the result process activities are organized in the pools and lanes according to the defined time granularity:

1. by days – then the first lane contains activities that are performed from the process starting date to next day
2. by weeks – the tool organizes activities in the lanes by the weeks.

It should be possible to calculate the end time of each activity – it is an elementary mathematical task, hence the start date and time of the process is defined and the processing time of each activity is known. The tool should calculate the end time or the start time of the whole process by summing up the processing and wait times for all activities and adding the result to process start date or subtracting the result from process end time respectively.

The first step of transformation is to create the base of the model – the lanes named according to the time periods of time granularity chosen by users of the model. Then the tool organizes activities in the created lanes according to the defined processing and wait time of the activity.
4.2 Transformation to business rule dimension

Although BPMN does not provide any specific construct for specifying rules, they are usually represented through the gateway logic [1]. To transform business process model from any dimension to business rule dimension it is required to store the details about the business rule in the process as a set of attributes of a gateway element. The attributes defining business rules are:

1. name of the business rule – the user defines it as the name of a gateway,
2. input of the business rule – the tool automatically identifies input as the artifact associated with the input flow of the gateway,
3. possible outputs of the business rule – the tool automatically identifies the outputs as the artifacts of the gateways output flows,
4. condition of the business rule – the user has to define the condition statement in the gateway properties, which could be as the names of the outgoing gateway flows.

The first step in transformation is to create the base of the model – the lanes are named according to the names of the business rules. Then the tool organizes activities in created lanes according to activity’s association with a particular business rule – whether activity’s flow is an input or an output of the gateway controlling the process flow in compliance with the business rule.

4.3 Transformation to information dimension

In BPMN a data objects provide information about what the process does – how documents, data, and other objects are used and updated during the process [3]. The name “data object” may imply not only an electronic document, it can be used to represent many different types of objects, both electronic and physical ones [3]. In general, data objects are defined as inputs of the activity and as the outputs of the activity using the association as the type of connection to activity or to input or output flow.

In the business process the same data object can be an input for different activities – in this case BPMN specification [3] offers to connect this data object with all activities that this data is associated to using association. However, adding more connecting objects to the model can make it almost unreadable. Moreover, it can be time consuming to follow all the leads from one data object to every associated activity. The solution is to transform business process model to information dimension – to group activities in the swimlanes by the information they are using in their inputs or producing as their outputs.

To transform business process model from any dimension to information dimension it is required to define inputs and outputs of every activity in the process using data object elements. Data object elements can be associated with the sequence flow – in this case the tool has to be able to identify whether the data object is an input or an output of activity, or the data object can be associated directly to the activity using input or output associations [3]. The data object element has to be refined by adding the name of the object and the tool should generate the unique identifier of the data object – to differentiate data objects with the same names.

The first step of transformation is to create the base of the model – the lanes are named according to the names of data objects used in the process. Then the tool organizes activities in created lanes according to activity’s association with a particular data object taking into consideration whether data object is an input or an output of the activity.

4.4 Transformation to performer dimension

If the original business process model is created in other than performer dimension, then it should be possible for the user to define in the properties of the activity the name of the performer which is responsible for the execution of the activity.

The logic of the transformation to performer’s dimension is described in previous sections – the tool groups activities in the created lanes by the performer.

5 Example of transformations

In this section a practical example of business process model transformation to time, business rules and information dimension is presented. The business process presented in this section defines the order of work for a testing team in the software development company. There are two roles in the testing team – the testing manager and the tester, each participant is responsible for carrying out a particular set of activities. The business process flow is controlled by four gateways, which are regarded in this article as business rules, and there are eight different artifacts produced and used by activities in the process.

Figure 1 describes the testing process from the performer’s dimension – all process activities are grouped in the lanes by performer, lanes are organized in one common pool – Testing team. The process model contains the artifacts produced and used by activities, however the process model does not reveal any
information about time factor in the process – it is not defined when process is initiated and how long does it take for each activity to execute it. The time perspective of the business process is one of the crucial factors in the bottleneck analysis of the process. One of the possible solutions to represent the time factor in the model is to define the time within each activity as part of the activity name – e.g. “Prepare testing plan, 2 days”, however this solution does not prove itself useful when the modeler decides to present the process activities performed by weeks, days or months. Such representation of process activities could be possible without redrawing the process model in the tool with dimension transformation functionality.

Figure 1. Business process according to the performer dimension

Figure 2 shows the draft of the process model with defined time dimension attribute after parameterizing BPMN process model elements. In this process model the start of the process is defined as a date, the processing time for each activity is defined in the “dd:mm:hh:mm:hh” format, the input flows show the start date of the activity, the output flows show the finish date of the activity.

Figure 2. Business process according to the performer dimension with defined time attributes

The process model in the figure reveals such important information about the time factor in the process – which process activity is the most time consuming, however the process model is still presented in the performer's dimension. To present the process model in time dimension, the user of the modeling tool has to invoke the transformation function and choose the time granularity in order to organize process activities in the lanes according to time dimension. represents the process model in time dimension by weeks.
Figure 3. Business process model according to the time dimension

The important detail is to keep activity’s performer name for process model enhancement. The model presented in organizes process activities by weeks and now it is obvious which set of activities are performed in the first, second or third week of the process.

represents the business process in business rules dimension – process activities are organized in the lanes by business rules.
Each lane contains one gateway or one business rule and activities which produce input for the gateway as well as activities which are initialized by business rule logic according to input data. The process model in business rules dimension presents which activities are affected by a particular business rule and what decisions are made during the process.

represents the business process in information dimension – process activities are organized by data objects that activities are using or producing. Overall there are 8 data objects used and produced in activities outputs, so the model in information dimension contains 8 lanes – each for one data object and contains activities to which data object is an input or an output. The business process representation in information dimensions shows by what set of activities a particular data object is used or produced.

Figure 4. Business process model according to the business rules dimension
6 Conclusions and future work

This paper introduces the foundations of business process multidimensional modeling using BPMN standard-based approach, a standard developed by the Object Management Group (OMG) [4]. A business process model is always modeled from a certain perspective of observation – e.g., time, business goals, performers, information and other perspectives [2]. Each perspective of observation may be regarded as a modeling dimension [2]. When observing business process model from certain dimension, the observer is getting all the details only from represented point of view – process activities are organized according to the certain dimension, however business process analyst has to view the process from different dimensions to discover bottlenecks and areas of potential improvement in a process, the most time consuming process activities and the process nodes that could be optimized. In this paper the following details of business process are regarded as business process modeling dimensions:

1. Performer dimension – represented in the business process model using lanes and pools

Figure 5. Business process model in the information dimension
2. Time dimension – time dimensions attributes are defined in the attributes of BPMN events and activities
3. Business rule dimension – business rules are represented using BPMN gateways
4. Information dimension – information units produced and used in the process are represented using data objects

To represent business process from the most important dimensions, with the purpose to reveal all the relevant details about the process, the modeling tool with dimension transformation functionality is needed. The modeling tool with built-in dimension transformation functionality is a crucial point in multidimensional modeling – manual transformation may create syntax and semantic errors and is not time efficient.

As for the notation for multidimensional business process modeling the paper comes forward with BPMN business process modeling notation. In BPMN Lanes are used to organize and categorize activities within a pool. The meaning of the Lanes is up to the modeler – BPMN does not specify the usage of lanes [3] therefore lanes can be used for organizing process activities in different dimensions.

Transformations represented in this paper were quite simple and straightforward, mainly requiring only regrouping of the elements. In more complex cases more sophisticated changes in the models are needed to transform them from one dimensions to another. The future work will address these problems as well as code generation from multidimensional business process models.

Acknowledgements

The author acknowledges Professor Dr. sc. ing. Marite Kirikova for valuable comments on the draft of the paper.

References

ENTERPRISE MANAGEMENT VIEW BASED SPECIFICATION OF BUSINESS COMPONENTS*

Edvinas Pakalnickas¹, Saulius Gudas¹,²

¹Kaunas University of Technology, Information Systems Department, Studentu St. 50, Kaunas, Lithuania, edvinas.pakalnickas@stud.ktu.lt
²Vilnius University, Kaunas Faculty of Humanities, Muitines St. 8, Kaunas, Lithuania, saulius.gudas@vukhf.lt

Abstract. Identification of the system components and their interfaces is one of the primary research problems in software development process. This paper describes domain-based identification of components and interfaces which are derived from enterprise management functions. The business domain related components and interfaces are specified using component-based system model, representing a business domain. The approach refines a set of rules aimed to gather software components from domain model. The resulting specification of enterprise management functions, related system components and interfaces is presented.

Keywords: Service-oriented design, business modeling, component-based systems engineering, business function.

1 Introduction

Modern organizations act in very dynamic business environments; therefore information systems supporting business needs should be flexible, extensible and maintainable. Everyday information systems of Enterprises grow in size and became more complex. In order to manage complexity is needed to break down system into subsystems with well defined interfaces. The subsystems are less complex than overall system and can be implemented as software components.

Interface-based design is evolution of object-oriented design [1][5]. Interfaces have made software design more adaptable to rapidly changing business environment. Using interfaces software systems achieves reusability, extensibility and maintainability. It is crucial to identify and to define signature of interfaces during early IS design stages. Correct specification of interface signature increases maintainability and reusability, because implementation can be separated from the public interface.

In order to model loosely coupled distributed systems of components, should be used component-based software engineering (CBSE) principles [4]. Main advantage of component-based approaches – loose coupling of components, services of components are reachable through interfaces, client of component doesn’t care about internals of component or how component is implemented.

For identification and specification of components and their interfaces is needed to use models, because using models is possible to reduce the complexity by translating business requirements into software implementations. Models became more important artifacts and they are valuable as much as source code [13].

Services-oriented architecture (SOA) is an architectural pattern based on interconnected services, which communicates only via messages through exposed interfaces. SOA is evolution of CBSE and provides basis for creation of loosely coupled distributed information systems [14].

Our goal is to present the approach for discovery of business components (BC), specification of Business Function Components (BFC) which can be used as construction blocks during Service-Oriented applications design. Also is presented BC’s and BFC’s interfaces discovery process. The rest of the paper is organized as follows. In section 2 is described current trends of SOA approaches and main ideas of proposed approach. In section 3 is presented enterprise management view and enterprise management function which is formal basis for specification of business function components. In section 4 is described how enterprise management function is mapped to business function component.

* The work is supported by Lithuanian State Science and Studies Foundation according to High Technology Development Program Project "VeTIS" (Reg.No. B-07042).
Section 5 presents relationships between business function, business function components, business components and software components. In section 6 is described purpose and properties of Component-based system model. Section 7 describes identification strategy of Business Components. Section 8 presents initial design principles of business function component. Section 9 depicts difference between CBSM layers and layered applications architecture. Finally, section 10 concludes the paper and discusses future work.

2 Related work

There are various approaches for development of service-oriented, component-based software and they are kindly diverse [3][4][5][7]. Recent work in the domain of service-oriented architecture provides a suitable technical foundation for making business processes accessible within enterprises. Services, from IT viewpoint, correspond to reusable software components used to implement activities within a business process [4][12].

Software components of complex enterprise application are designed and implemented across multiple levels of abstraction [4][8]. For example in 3-layers applications architecture, components in one layer can interact only with components at the same level or components from lower levels. In presented approach business components and their interfaces are identified using created component-based system model (CBSM) [4][17][18][20], are designed at the same level of abstraction, grouped by different domains of enterprise activities [16] and joined by different types of interfaces. Such approach allows creating design of loose coupled system and increases flexibility, reusability and reliability of designed system.

The service-oriented paradigm offers a flexible approach for reusability of existing business processes and creativity/modeling of new ones. It allows dynamically building business processes by composing existing services tailored to enterprise needs. However existing SOA approaches are lacking of clear definition how to identify and to specify signatures of interfaces, which would enable faster and more reliable service or components composition process. This work increases the need to consider a service-based modeling of business components from enterprise management point of view [10]. Next section shortly defines how the enterprise management view is used in our approach for analysis of Enterprise activities.

3 Enterprise management view

The analysis of the Enterprise as a hierarchy of information processing activities from the management point of view is presented in [11] as formalized model of the Enterprise management. The information flow at the Function-Process interaction is a background of the formalized model of the Enterprise management model and is described in [10]. The major component of the Enterprise management model is an Enterprise Management Function.

A management Function in [11] consists of the predefined sequence of mandatory steps of information transformations (Interpretation, Data Processing, Decision Making, Realization); these steps compose a management cycle (an information feedback loop). An enterprise management Function F(i) is initiated by some Event – a fact or a message associated with some internal or external (environmental) object. A definite set of attributes (a set of information items) is formed and transmitted during implementation of management Function F(i) at the each management cycle step. The paper [10] presents more detailed content of Function F(i) since it defines a sequence of definite types of interacting information activities (Interpretation, Information Processing \{Data Processing, Decision Making\}, Realization) directed to control enterprise Process P(j) (Figure 1).

![Figure 1. Information interactions of enterprise management function F(i) and enterprise process P(j).](image-url)
Figure 1 presents the structured model of the Function-Process interaction. The concept Process is assumed as “a black box”. The internal structure of Process is not important, only the information about the state of Process is considered from the point of view of some definite management Function. The concept Process is characterized by a set of Process state attributes (this set comprises subsets of Input flow attributes, Output flow attributes and Process attributes) and it is influenced by the output of management Function – a set of Process control attributes.

It is assumed, that Process and management Function are activated by some Event. A definite set of state attributes of an activated Process is the information flow defined as an input of (one or more) specific management Function.

So, from the management point of view an Enterprise model (at business level) comprises of a hierarchy of well defined Enterprise management functions (see Figure 1.) with predefined structure. The model-driven development of business applications requires keeping the logical relationships between an Enterprise model and Information System model.

4 Structure of Enterprise management function at Information System level

Enterprise management function at software level should expose the logic of information transformation for all management cycle phases (Interpretation, Information Processing {Data Processing, Decision Making}, Realization), therefore management functions at information system level should be accessible through interfaces. If we look at management functions from applications services viewpoint it is clear that we should identify connections of functions. Identification of function’s connections to other functions and defining of interrelated system helps to understand which and how functions can be joined, this is a crucial step for improving architecture of information system.

Discovering interfaces of enterprise management functions may be as valuable as defining the functions, because it is possible to manipulate and manage the connections through interfaces, while the functions remain essentially unchanged black boxes.

In presented approach, at information system level, enterprise management functions will be mapped to business function components (BFC). Business function components will expose interfaces, which will be used to provide input attributes and get output attributes of management function. It was defined following types of interfaces for business function components:

- **BDI** – interface will be used for communication with Decision making part of management function, at information system level this interface will be used for interaction with user interface.
- **IPDI** – interface will be used for communication with Interpretation part of management function, this interface will be used for interaction with computational resources of IT infrastructure.
- **IDI** – interface will be used for communication with Data Processing part of management function, at information system level this interface will be used for interaction with data resources.
- **PTDI** - interface will be used for communication with Realization part of management function, at information system level this interface will be used for interaction with technological resources of enterprise.

5 Business functions, services, components

Business functions are more are more abstract and therefore are more difficult to deal with them. In order to reduce level of abstraction in our approach we use enterprise management functions, because they have defined structure and it is possible to turn them into software services that enterprise needs in order to reach its goals.

A business service is a function of the business that is offered to internal or external customers. SOA service provides the ability to share business functions by providing information, calculations or data processing; therefore enterprise management functions can be used as basis for definition of interfaces provided by system.

In our approach services of information system are structured around the concepts of enterprise management functions, business processes and business components. These concepts are addressed through a number of components, entities and interfaces in the business layer and after that they are mapped to IT infrastructure layer.

Business components (BC)[4][18] encapsulate the business process or entity. By using business components an enterprise management function consists of number business components that collaborate to deliver functionality to the users of system. Business component provides services to outside via interface, these services can be implemented to one or several enterprise management functions within enterprise. Business components provide
business related services through interfaces to enterprise management functions, business components are implemented as software components.

**Business function component (BFC)** is software module that implements semantics of enterprise management function.

Figure 2 depicts relations between enterprise management function, business function component and business component. Business function component and business component will be implemented as software components.

![Figure 2. Relations between management function and components](image)

6 Component-based system model

We have created new model – component-based system model for identification of business components (BC) and their interfaces. This model implements requirements of Information Architecture Framework for Enterprise Integration [16].

CBSM provides four separate domains for the specification of enterprise management functions and provides framework for identification business components and their interfaces:

**Business domain (BD)** - includes business processes, employees, their roles and responsibilities, organizational structure, workplaces. It includes applications for management control and strategic planning.

**Information processing domain (IPD)** - includes algorithms for calculations, business rules and their interrelationships.

**Information domain (ID)** - includes business data and knowledge, usage of it, ownership, distribution and composition.

**Product technology domain (PTD)** - includes technological processes and facilities for development of the enterprise products.

Main purposes of CBSM:

- To help identify business components (BC) and their interfaces
- To store knowledge about business components and share it across enterprise
- To bring component and service way of thinking to higher level of abstraction – during the initial phases of information system life cycle.
Figure 3. Component-based system model

Figure 3 represents component-based system model. Components are displayed as rounded rectangular, interconnected with interfaces.

CBSM component specification can be described as:

\[ C_i = \{ A, I(S_1, S_2, S_3, S_4, S_5, S_6) \} \]

Where:
- \( C_i \) - component belonging to one of CBSM domains;
- \( A \) - attributes of component \( C_i \);
- \( I \) - component’s interfaces for connection with other components.

Each CBSM component has different set of interfaces:

- **Business domain components (BDC)** -
  \[ S_{BDC} = \{ S_1, S_2, S_3 \} \]

- **Information processing domain components (IPDC)** -
  \[ S_{IPDC} = \{ S_2, S_4, S_6 \} \]

- **Information domain components (IDC)** -
  \[ S_{IDC} = \{ S_1, S_4, S_5 \} \]

- **Product technology domain components (PTDC)** -
  \[ S_{PTDC} = \{ S_3, S_5, S_6 \} \]

Figure 4 illustrates four types of CBMS components and their interfaces.

Each pair of domains communicates through different type of interface (see Figure 5a)) or each type of interface joins components from two different domains (see Figure 5b)). Component-based system model allows...
specifying network of interconnected components; for example interface $S_2$ joins component IPDC with components BDC and BDC2, interface $S_1$ joins component BDC with components IDC and IDC2.

![Diagram of CBMS components and interfaces]

Figure 5. CBMS components and interfaces. a) set of different interfaces; b) example of interconnected components

7 Business components identification strategy

CBSM allows to create specification of enterprise management functions; for example order processing, procurement of materials.

![Workflow of order processing management function]

Figure 6. Workflow of order processing management function

Figure 6 presents workflow model of order processing enterprise management function represented in BPMN notation [15]. Designer should analyze workflow model of enterprise management function and identify business components and appropriate set of component’s interfaces. Many changes of component’s interfaces impact other components and result instability. Due to this it is crucial from beginning of system design identify main interfaces which will be constant during the life cycle of information system.

Figure 7 represents the algorithm for business component identification.
Designer should apply the following rules, which can help to identify business components and interfaces between them:

- the computational processes should be transformed into the components of the information processing domain;
- the management processes and the gateways should be transformed into the components of the business processes domain;
- the information flows connecting processes in the workflow model should be transformed into the information domain components;
- processes related to materials should be transformed into the components of the technological processes domain;
- identified components should be joined with interfaces S1-S6, interface number depends on components place in CBSM (see Figure 5 a)).

Figure 8 represents the example of component-based system model of order processing enterprise management function.
8 Design of Business Function component

When CBSM is created, designer can start design of business function component. Component-based system model captures main business components for particular enterprise management function. It is possible to transform business components from CBSM to the technical design model of BFC. Each business function component will provide four types of interfaces (see Figure 9) and will be implemented as application component or application service:

**BD components:** each component of business process domain is mapped to a method in BDI interface of business function component; these components become business services provided by the information system.

**IPD components:** each component of information processing domain is mapped to an operation of IPDI interface; components from IPD layer become computational services provided by the component.

**ID components:** each information domain component is transformed to data access operations, which communicates through interface IDI.

**PTD components:** each product technology component is mapped to an operation of interface PTDI; these components become output/input for technological environment of enterprise.

Business function component specification is described as:

$$BFC_i = \{A, I(BDI, IPDI, IDI, PTDI)\}$$

where:

- $A$ - attributes of component $BFC_i$;
- $BDI = \{BDC_1, ..., BDC_j\}$ - interface will have set of methods to invoke functionality of BD components;
- $IPDI = \{IPDI_1, ..., IPDI_j\}$ - interface will have set of methods to invoke functionality IPD components;
- $IDI = \{IDI_1, ..., IDI_j\}$ - interface will have set of methods to invoke functionality of IPD components;
- $PTDI = \{PTDI_1, ..., PTDI_j\}$ - interface will have set of methods to invoke functionality PTD components;

Figure 10 depicts discovered interfaces of Order processing business function component, component has definition of interfaces and after detailed design can be implemented as software component.
In process of business function component F1 design it is possible to find out that some of business components inside BFC1 component should access services from management function F2 component BFC2. In such case business components use interface of his domain in order to reach functionality of other components; for example PTDC1 component will use PTDI interface in order to access BDI interface of BFC2 which implements management function of technological process.

9 CBSM layers and layered enterprise applications

Layered application design is applied extensively in the software development process. A common implementation of enterprise applications is three-layered application. It is defined three main layers - presentation, business, and data. These three layers are almost always needed for enterprise applications.

CBSM domains must be not confused with layered application design; domains provide conceptual structure for business components identification at the enterprise level. Main difference between CBSM domains and layers of enterprise applications – CBSM has four logical business layers – domains, domains can be joined horizontally – components from different domains can use services from other domains components (see Figure 10). In case of n-tier layered application architecture – system has a number of logical layers and those layers are joined vertically, where layer n is above layer n+1, layer n can use services of layer n+1.

10 Conclusions and future work

Enterprise management functions provides the right level of granularity for a business description of the enterprise operations by encapsulating number of activities in an enterprise and interactions that occur among the different business processes and entities, proposed approach enables to keep the logical relationships between an Enterprise model and Information System model.

Discovery and modeling of components is analyzed from the new perspective named as management point of view. It enables to discover interfaces of enterprise management functions and provides basis for implementation of enterprise management functions as software components, which can be used for composition of service-oriented information systems.

With component-based system model it is possible to identify business components and define interfaces between them, ensuring integrity of particular enterprise management function.
Business components identification and implementation as application components enables business requirement traceability to IT infrastructure.

The future work has two directions: business function components and business components are part of Computation-Independent Model and they can be automatically transformed into initial Platform-Independent model. Therefore will be developed MDA-style transformations [13] for proposed approach; another direction – to create method for detailed business function components and business components specification applying UML-intensive framework [2] and providing an engine for controlling the business function components composition.

References