

Representing Variability in Enterprise Architecture

A Case Study

Nino Rurua · Rik Eshuis · Maryam Razavian

Received: 17 August 2016 / Accepted: 11 May 2017 / Published online: 7 December 2017
© The Author(s) 2017. This article is an open access publication

Abstract Organizations that operate on an international scale have a high variation of business operations, caused by country-specific regulations and compliance requirements. The differences in requirements lead to variability in the designed business processes and their supporting applications and infrastructure technology. Such variability should be represented in enterprise architectures, which are structures that align business operations to IT. However, current approaches to enterprise architecture are agnostic to variability. The paper presents an explorative case study, performed at an international high-tech company in the area of electronic invoicing, in which a solution for representing variability in enterprise architecture is designed. The developed solution has been validated by company experts.

Keywords Enterprise architecture · Variability modeling · ArchiMate

Accepted after two revisions by Prof. Dr. Zdravkovic.

Electronic supplementary material The online version of this article (<https://doi.org/10.1007/s12599-017-0511-3>) contains supplementary material, which is available to authorized users.

N. Rurua
Architecture, Strategy and Performance, Philips Electronics
Nederland B.V, High Tech Campus, 5656 AE Eindhoven, The
Netherlands
e-mail: nino.rurua@philips.com

Dr. R. Eshuis (✉) · Dr. M. Razavian
School of Industrial Engineering, Eindhoven University of
Technology, P.O. Box 513, 5600 MB Eindhoven, The
Netherlands
e-mail: h.eshuis@tue.nl

Dr. M. Razavian
e-mail: m.razavian@tue.nl

1 Introduction

Today's dynamic business environment brings both benefits and challenges to large multi-national organizations. Globalization forces organizations to deal with different industry standards, government regulations and compliance requirements, causing variations in business data, business processes and software systems. Identifying and handling such variations is crucial for organizations to manage their business operations in an efficient way.

Organizations use enterprise architecture (EA) as an instrument to align their business operations and IT systems. An EA consists of different modeling artifacts that together describe an organization's strategic goals and its operations, including business processes, and supporting IT systems (Lankhorst et al. 2013; Winter and Fischer 2007). Thus, an EA manages the complexity of enterprise wide IT systems with respect to business goals to improve Business-IT alignment (Winter et al. 2010).

The complexity in design and architecture of enterprise systems is often due to variation in the organization's business requirements, products or services, as well as implementation options. However, an approach for managing variability in EA is lacking. Variability management has been extensively studied in the literature for subjects related to EA, such as Software Product Line Engineering (SPLE) (Pohl et al. 2005; Sinnema and Deelstra 2007; Svahnberg et al. 2005) and Business Process Management (BPM) (Rosemann and van der Aalst 2007; Hallerbach et al. 2010), and service-oriented systems (Galster and Avgeriou 2015; Park et al. 2011). However, EA artifacts have to deal with more complex environments than software or service artifacts.

This paper presents an explorative case study in which a solution for representing variability in enterprise

architecture (EA) is developed. The case study object is the electronic invoicing process for Latin America countries of Philips, a high-tech company that operates on a worldwide scale. Each country in this area has stated or is stating its own fiscal requirements pertaining to e-invoicing, leading to country-specific business processes and EA artifacts, whereas the company aims to standardize the processes as much as possible to gain the most optimal performance. We design a solution for representing variability in EA by extending the EA metamodel used within Philips with variability management concepts taken from the literature. We use the extended EA model to embed variability in the EA artifacts that relate to electronic invoicing. The solution design has been validated by company experts.

As a first step to improve the lack of support for managing variability in EA, the variability representation developed in this paper allows to succinctly represent in EA different IT solutions as well as the various business requirements they fulfill. This allows different stakeholders from business and IT to understand the used IT solutions for a business scenario (e.g., invoicing in different countries), which in turn helps them to identify opportunities for reuse and application of existing, proven IT solutions. This way, the solution helps decision makers to align business requirements and existing IT solutions in a more effective way, which is in line with the main goal of EA.

The remainder of this paper is organized as follows. Section 2 discusses related work on EA and variability management. Section 3 presents the research approach, which is based on design science. Section 4 presents the electronic invoicing process for Philips in Latin America and analyzes its variability. Section 5 presents the solution design, which is an extension of the EA metamodel that is currently used. Section 6 presents an evaluation of the solution by company experts. Section 7 ends the paper with conclusions.

2 Related Work

We discuss related work in the field of enterprise architecture and current approaches of variability management in information and software systems.

2.1 Enterprise Architecture

Enterprise architecture (EA) is a coherent set of principles, methods, and models that are used in the design and realization of the organizational structure, business processes, information systems, and infrastructure of an enterprise (Lankhorst et al. 2013; Winter and Fischer 2007; Winter et al. 2010). For instance, the EA framework by Zachman (1987) describes an organizational structure from various

stakeholder perspectives and provides support for business, information systems (IS) and information technology (IT) alignment.

The Open Group Architecture Framework (TOGAF) (The Open Group 2011) is an industrial standard for developing EAs. TOGAF uses ArchiMate (The Open Group 2013) as modeling language for EA. ArchiMate was developed to provide a uniform representation for diagrams that describes the Business, Information Systems, Technology layer of an EA. Recently, a layer for Motivation has been added to represent intentions and their sources (Engelsman et al. 2011). One of the major advantages of ArchiMate is that it provides notations for relationships between concepts on different layers. These relationships make it easier to show and trace dependencies between objects across EA. All the relevant concepts and relationships are organized in an EA *metamodel* which defines the underlying language of a framework or methodology. Another language is BPMN to model business processes in the Business architecture layer (Object Management Group 2011). Philips has selected TOGAF as EA approach and ArchiMate and BPMN as main modeling languages for EA. We therefore consider these two languages in the remainder of this paper.

In recent years several research works have investigated how to combine ArchiMate, e.g. with resource modeling (Azevedo et al. 2015), value modeling (de Kinderen et al. 2014), business models (Iacob et al. 2014), access control (Korman et al. 2016), and simulation (Manzur et al. 2015). However, there are no works that study variability management for ArchiMate models.

To the best of our knowledge, there is no related work on representing variability in EA. There is some related work on flexibility, modifiability, adaptability of EA, which are external quality attributes that deal with changing enterprise systems, whereas variability addresses managing variation inside an enterprise system. Lagerström et al. (2010) define an EA metamodel for analyzing the cost of modifying enterprise systems. Mikaelian et al. (2011) define an approach for managing uncertainty in EA based on real options analysis. Flexibility and adaptability mean how well an EA deals with changing environments (Erol et al. 2010; Wilkinson 2006). None of these papers develops support for representing or managing variability in an EA. Variability is used implicitly in enterprise modeling approaches that take context into account, which naturally leads to variability (Bērziša et al. 2016).

2.2 Variability Management

Variability management has been studied in the area of Software Product Line Engineering (SPLE) (Chen and Babar 2011; Pohl et al. 2005; Schmid and Isabel 2004;

Svahnberg et al. 2005; Sinnema and Deelstra 2007). Variability in SPLE is the ability of a software artifact to be extended, changed, customized or configured depending on the specific context (Sinnema and Deelstra 2007). An extensive survey of variability modeling for SPLE is provided by Czarnecki et al. (2012).

Variability is also studied in the area of Business Process Management (BPM) (Rosemann and van der Aalst 2007; La Rosa et al. 2009; Hallerbach et al. 2010). While SPLE approaches mostly deal with modeling features and ways to combine and implement them, in BPM variability management is about efficiently handling different variants of a business process. Ayora et al. (2015) provide an extensive review of different variability approaches in BPM.

Overall, there can be two major categories distinguished in variability management techniques from the modeling perspective. The first group integrates common and variable parts into a single model, while the second one explicitly represents variability in a separate model, resulting in multiple models (Ayora et al. 2015; Pohl et al. 2005).

Understanding and explicitly documenting variability improves decision-making, communication between stakeholders and traceability between variation causes and effects (Pohl et al. 2005; Czarnecki et al. 2012). In order to adequately describe and document variability, Pohl et al. (2005) suggest answering four questions about software families: what varies, why, how, and for who? These questions relate to the common terms used to express variability in literature (Milani et al. 2016; Schmid and Isabel 2004; Svahnberg et al. 2005; Ayora et al. 2015). However, the list proposed by Pohl et al. (2005) does not include a question regarding dependencies or relationships between variation points (where variation occurs) and variants. These elements play an important role in allowing traceability of variation (Schmid and Isabel 2004; Sinnema et al. 2006) in SPLE and processes. The solution for representing variability in EA developed in Sect. 5 is based on the variability concepts described by Pohl et al. (2005) but explicitly addresses traceability of variation.

3 Research Method

The case study object is the e-invoicing process in Latin America, each country in this region being a unit of analysis. The objective of the case study is to design a means to represent variability in the EA for these e-invoicing processes. To reach this objective, we have selected a design-science approach (Hevner et al. 2004) as research method, because of its relevance to the domain of information systems (IS) and the applicability of the approach to the

problem. The designed solution is an extension of the EA metamodel and EA artifacts for electronic invoicing in which variability, caused by differences in the compliance requirements, is embedded. These artifacts instantiate the extended metamodel. The research process followed an iterative cycle of developing and evaluating the solution, taking into account the business needs (environment) as well as the knowledge base.

The design process was divided into three major phases: to *Investigate and Analyze* the existing problem, environment and available resources, to *Develop/Build* the possible solution and to *Justify/Evaluate* the relevance and applicability of the proposed solution.

During the *Investigate and Analyze* phase, we searched for journal and conference papers from the past 10 years on variability management for enterprise, information systems, and software architecture and selected the most relevant papers based on the abstract, which were 17 journal and 6 conference papers. A condensed survey has been presented in Sect. 2. Next, we studied the existing EA used within Philips by interviewing two business architects and two IT architects and analyzing existing documentation. In parallel, we have analyzed the e-invoicing case study to explore the problems and issues in managing variability in EA.

We collected data for the case study using a direct method and independent analysis (Runeson and Höst 2009). The information from the stakeholders was obtained by attending multi-disciplinary team meetings and discussions. The team meetings were held to guide the design and implementation of e-invoicing processes in different countries. The goal of the stakeholders is to optimize and standardize the current and future e-invoicing processes of Philips in the Latin-American countries as much as possible. So far, Philips had implemented its e-invoicing processes in half of the countries in the area. The multi-disciplinary team meetings were attended by the business process owner, business process expert, IT architect, and business analyst for integration. In addition, we held semi-structured interviews with the process owner and process expert to acquire more information on the e-invoicing process. We also studied internal documentation within Philips on this topic.

In addition, information about the relevant fiscal regulations was studied using online resources such as white papers provided by third party service providers for tax/e-invoicing solutions. Due to a language barrier, official government websites could not be used, but webinars offered a good opportunity to interact with the experts in the area and receive more country-specific details. The obtained information from the online sources was checked for relevance and accuracy with the local process expert of Philips in the region.

For the *Develop/Build* phase, the Philips case has been used to identify and explore variation in the requirements that affect decisions during process design. In addition to that, we derived a solution design by first defining variability-related concepts, such as variation drivers, variation points and dependencies and incorporating these concepts in the EA metamodel used within Philips. Next, we developed EA artifacts pertaining to the e-invoicing case. The artifacts instantiate the extended EA metamodel. The variability caused by different regulations is embedded inside the artifacts. We developed the solution in an iterative fashion by asking the primary stakeholders (field experts, IT and business architects) for feedback on the different parts of developed solution and refining the solution based on their feedback.

During the *Evaluation* phase, we checked the feasibility of the solution for managing the differences in the compliance requirements during EA modeling. Since our research area addresses a particular organizational problem, the solution was evaluated with respect to its utility within the practical context (Hevner et al. 2004). The utility of the proposed approach was assessed by experts who were not involved in the analysis and solution design phases. More details are provided in Sect. 6.1 and the appendix.

4 Analysis of Variability in E-invoicing

As first step in the design process (investigate and analyze), we have analyzed the business processes of electronic invoicing for countries in Latin America as well as the existing EA. We first explain the various requirements,

then the structure of the existing EA, and finally we analyze which different artifacts of the EA are affected by the variability caused by the different requirements. The design in the next section builds upon this analysis.

4.1 E-invoicing Process Requirements

One of the new developments in fiscal legislations worldwide is the adoption of electronic forms of invoicing. Digitalization of invoices and billing documents helps companies save costs by providing more control and automation of the process, and by eliminating the inefficiency caused by manual processing of paper-based invoices (Koch 2014). However, switching to new ways of invoicing is related to significant changes in the organizational processes, and requires alignment with not only internal processes, but also third-party solution providers.

The trend of mandating electronic invoicing (e-invoicing) and fiscal reporting has been increasing in the countries in South America in the past few years (Economist 2014). Starting with the government of Brazil, the procedure has been regulated in for instance Argentina, Mexico, and Chile. Even though each authority intends to have an effective means for tax collection, the requirements still differ in multiple aspects that affect the generic process flow of e-invoicing (Economist 2014).

The requirements regarding fiscal regulations have been elicited from global and local process experts in finance; they are not stable due to changing legislation. The requirements are presented in the form of a matrix with corresponding countries as columns in Table 1. Each row cell is filled in with either Yes/No, indicating the existence of the requirement per country, or with a certain value,

Table 1 Fiscal regulations affecting e-invoicing process in Latin American countries

Requirement Class	Fiscal requirements	Argentina	Brazil	Chile	Colombia	Mexico	Peru	Uruguay
Authenticity and integrity	FR1: Invoice must be approved by the gov	Yes	Yes	No	No	Yes	Yes	Yes
	FR2: Government prescribes invoice delivery method	EDI	EDI	EDI	EDI	EDI	EDI	EDI
	FR3: Invoice issuer must to be certified	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	FR4: Invoice must to be digitally signed	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Logistics	FR5: Goods must be accompanied with a doc. containing invoice authorization number	No	Yes	No	No	No	No	No
Invoice structure and content	FR6: Invoice must follow government prescribed format	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	FR7: Invoice must include pre-defined number range to be present provided by the government	Yes	No	Yes	No	Yes	No	Yes
	FR8: Invoice must be in government-specified language	SPA	POR	SPA	SPA	SPA	SPA	SPA
Customer Invoice	FR9: Invoice sent to the customer must include protocol number	Yes	Yes	No	Yes	Yes	No	Yes
Archiving	FR10: Invoices must be available online for download (government-specified number of months/years)	Yes	Yes	Yes	Yes	Yes	Yes	Yes

such as language or data type. The red color indicates differences or deviations from the general pattern in the row, marked in black.

Because of either the prolonged process of eliciting fiscal regulations via indirect communication with stakeholders, or still lack of official and clear information from respective governments, there is no definite information concerning some requirements. These requirements are marked in italics. In certain countries, like Columbia and Uruguay, it is expected that the legislation on mandating e-invoicing will be effective in the following years. To get a complete set of requirements, we assumed the most likely choices based on consultation with the domain architect about the current practice and expected legislations.

4.2 Enterprise Architecture Metamodel

Philips uses EA to ensure more efficient business operations by clarifying the strategic context and business capabilities of the organization. Philips has created EA models for documentation and communication purposes of the baseline and target IT landscape. In addition, the models have been enhanced with stakeholder requirements/concerns to analyze the landscape for better decision-making.

The enterprise architecture metamodel of Philips is based on ArchiMate and has four layers: Motivational, Business, Application and Technology. Figure 1 shows an abstract version of the metamodel. To give an indication of the complexity, Philips has 10 L1 processes and around 300 L3 processes and 3000 software systems. Figure 1 omits elements and relationships about the internal way of working of Philips that are of strategic importance. However, the omitted elements do not impact the parts of the metamodel that are relevant for this research, which is the distinction between different abstraction levels in the business layer and the link between Motivational and Business layer. The parts in orange are extensions related to representing variability and discussed in Sect. 5.

The Motivation layer describes the highest level of organization goals. Using ArchiMate objects, such as the goal, driver, and requirements, this part shows the underlying motivation for design or change of EA components. A goal is an end state that a stakeholder intends to achieve. A driver is something that creates, motivates, and fuels a change in an organization.

The Business layer describes the business processes at seven different levels of granularity. The highest level, L1, describes the area of the domain (i.e. Finance), L2 describes a Process Group (i.e. order-to-fulfillment), L3 specifies a standalone unique process (i.e. Manage Sales Order), L4 describes reusable pieces of flow in the processes, L5 describes tasks done by a single person at a time,

L6 specifies a further breakdown of tasks, and L7 provide detailed guidelines on how to accomplish a specific piece of work. A process is executed by a business role, which is played by people (not shown) or software systems (Technology).

The Application layer specifies the application support of business processes. An application component can be used to model any structural entity, such as software application, sub-application or information systems. At the application layer also interfaces and information flow channels between different components are defined.

The Technology layer specifies supporting software systems that realize applications. For the same application component there can be different software solutions available, which can be distinguished using the Location element.

4.3 Variation Points

Next, we analyzed how the listed process requirements and constraints affected the enterprise architecture, i.e., we analyzed the variation points in the EA artifacts such as process or application components, where the variation occurs (Schmid and Isabel 2004; Ayora et al. 2015; Galster and Avgeriou 2015). The variation points can occur in the Business, Application and Technology layers. As the lower layers represent realization of upper layers of the EA, variation points on one level indicate how the choice of one variant in an upper layer may affect its possible realization options in lower layers.

During the meetings with the stakeholders, i.e., the business process owner, business process expert and IT architect, we discussed and identified points in the architecture where there was more than one option or alternative for an artifact. The discussions to analyze variation options started with requirements and business needs. Later meetings to discuss business and IT aspects were held in parallel to ensure that the information from both sides were consistent and aligned with each other. For the reporting purposes, we follow the EA realization levels from top to bottom and start with the description of differences in the business layer followed by the implementation possibilities.

4.3.1 Business Architecture Layer

Given the base process, the fiscal requirements and the variability matrix (see Table 1), we identified branching points in the generic process and added new tasks to reflect government-involvement in the process. Following Milani et al. (2016), we considered two types of branching points: a *variation point* that splits the process into process variants, and a *decision point* that routes the process flow.

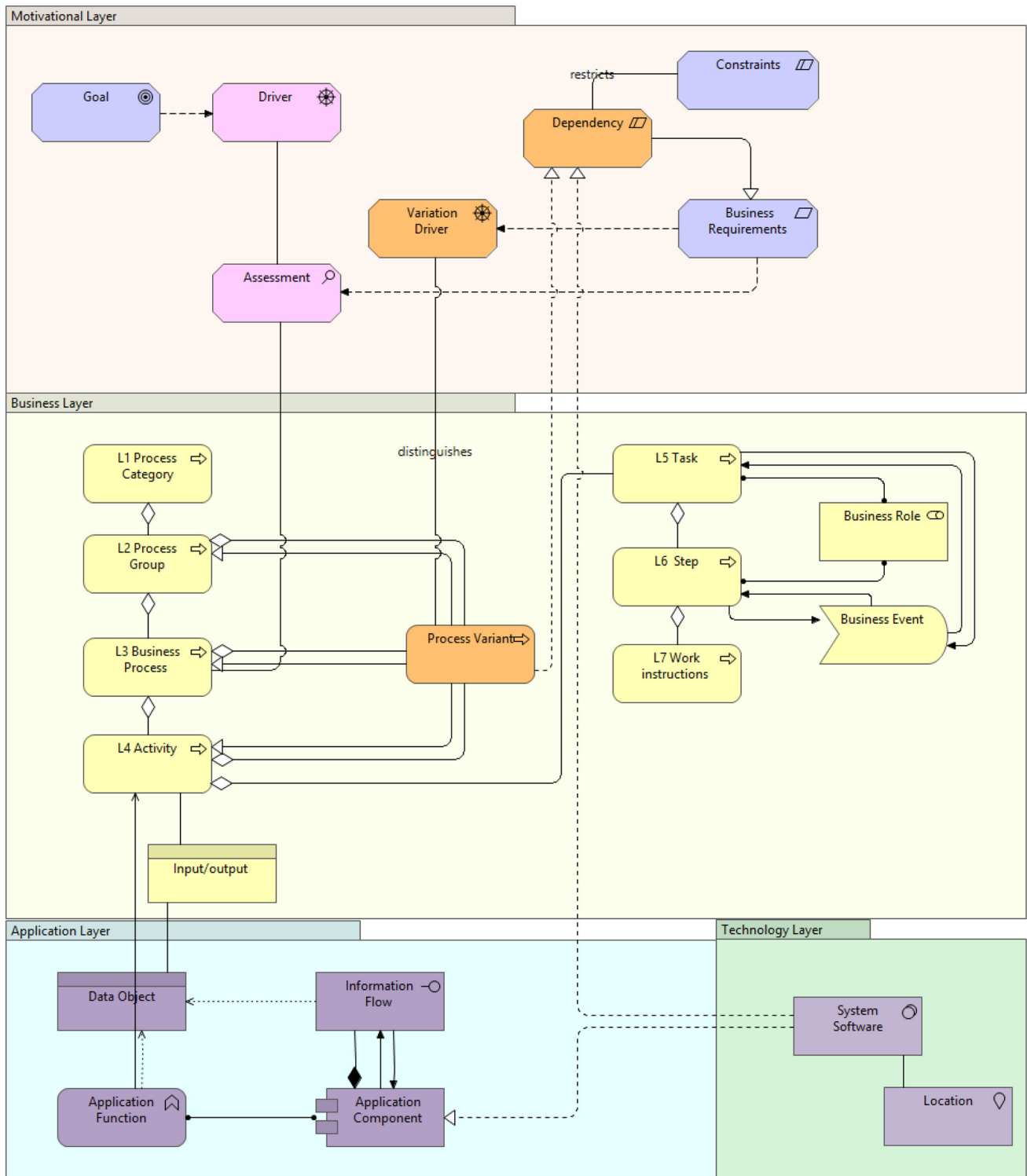


Fig. 1 Enterprise architecture metamodel and its variability-related extensions (in orange)

Together with the business process owner and expert, we identified three variants that occur in the sub process of *Transmit billing documents to customer*. Depending on the

fiscal regulations, an invoice issuer can be obliged to get the invoice approved before shipping the goods, and/or can be required to send only the approved invoice to the

customer. These variants are caused by the differences in the control flow of the process. Other types of variation occur due to differences in the business objects that are exchanged or that trigger the start of the process.

4.3.2 Application/Technology Architecture Layer

The variants in the business processes trigger variation in the application and technology layer due to specific requirements or options for their realization, for instance different ways of supporting the activities and different versions of the technologies that implement the process. Note that at these lower layers also other variations occur that are not caused by variation in processes, for instance differences regarding currencies.

The mapping of the identified variation points to the respective business and IT architecture objects in the architecture are presented in Table 2. In the column Variation option, each entry is a variation option, different values for each entry are alternatives.

We have also identified constraints that limit the e-invoicing process-related flow and the realization of identified variants in lower layers mainly due to the integration

of the systems of Philips and the governments. Example constraints are that the shipment authorization process requires synchronous information flow to exchange messages with the government system, and that synchronous information flow requires the use of cockpits for status monitoring. Due to space limitations we do not present the complete list of constraints here.

In this section, we only analyzed where variability in the EA should occur due to different process requirements. In the next section, we show how the identified variability can actually be represented in the EA. We use the identified variation points and options from the case study to build the relevant architectural models for the case study.

5 Solution Design

The solution design consists of two parts. We first present the extended EA metamodel that we developed to represent variability management concepts. Then we show sample architectural models that instantiate the extended metamodel. We developed these sample models for the e-invoicing case study to meet the different requirements

Table 2 Variation options mapped to corresponding EA artifacts across architecture layers

EA Layer	Variation Points	Variation Options
Business architecture, L3 process	1.1.1 Invoice customer	18.2.2a Invoice Customer (standard) 18.2.2b Invoice Customer (regulated)
Business architecture, L4 activity	1.1.1.2 Transmit billing data to customer 1.1.1.2a Shipment authorization is required 1.1.1.2b Invoice approval is required 1.1.1.2c The government requires a copy of invoice	{required, not required}
Business architecture, Data	L5: Determine invoice distribution method	{printed, PDF, EDI}
Business architecture, Event	L4: Generate customer billing data	{Billing run scheduled, Goods accompanying invoice required}
Application architecture, Data object	Business object: Input/output	{Gov.-prescribed format, non-gov.-prescribed format} {SPA, POR} Digitally signed
Application architecture, information flow	Information flow: communication	{Synchronous, asynchronous}
Application architecture	Application	{Cockpits, fire-and-forget}
Technology layer	Component	{ERP + in-house dev., outsource}
Technology layer	Component	{Any certified service provider, gov. specified service provider, certified supplier}

identified in the previous section. We developed the solution in an iterative fashion by asking the primary stakeholders for feedback on the developed artifacts; in this section we show the final version.

5.1 Extended Enterprise Architecture Metamodel

The EA metamodel used by Philips does not have concepts for variability management. Consequently, incorporating the variants identified in Sect. 4 in the EA results in EA artifacts that have a lot of overlap, which leads to higher costs in maintaining the architecture. Therefore, we extended the EA metamodel such that variability can be presented explicitly. There are two alternative scenarios for extending a metamodel: to enhance it with additional concepts from the same domain as the original concepts, or to augment it with new concepts from a different domain than the original concepts (Atkinson et al. 2015). In this case, we augmented the company metamodel with concepts for variability management.

Next, there are several, complementary mechanisms to actually extend a metamodel: using built-in extensions mechanism, metamodel customization, and model annotation (Atkinson et al. 2015). For augmenting a metamodel, the most effective way to make extensions is to use built-in extension mechanisms and model annotation (Atkinson et al. 2015). We have used both mechanisms.

Figure 2 shows the conceptual model of variability management we employ in this paper. The elements with

orange color indicate the extended elements that are added to the metamodel in Fig. 1 using the built-in extension mechanism of ArchiMate. The elements with yellow color describe the rationale behind the variability decisions; they are embedded in the metamodel by means of stereotypes within the business, application, and technology layers.

Variation drivers specify the causes of variability (Milani et al. 2016). Understanding and categorizing variants according to variation drivers can help in making design decisions in architecture. Variation drivers can be categorized according to the context in which they are used by linking interrogatives to organizational concepts (Milani et al. 2016): variability can depend on operational processes (*how*) through which a company delivers a product/service (*what*) to the market (*where*) for a customer (*who*) at a specific time (*when*) to meet a certain demand in the business environment (*why*). We added a variation driver in the conceptual model to specify the categorization of causes of variation that affect processes.

To describe variability itself, we use the key variability concepts defined by the Orthogonal Variability Model (Pohl et al. 2005): variation point, variation option and dependency. Variation points represent locations in the model where a selection or choice between options has to be made. Variation options can be optional, mandatory or alternative. A variability matrix is a document that relates existing alternatives to the corresponding variation points (cf. Table 1). The relationships between variation points and variants are constrained by dependencies. The

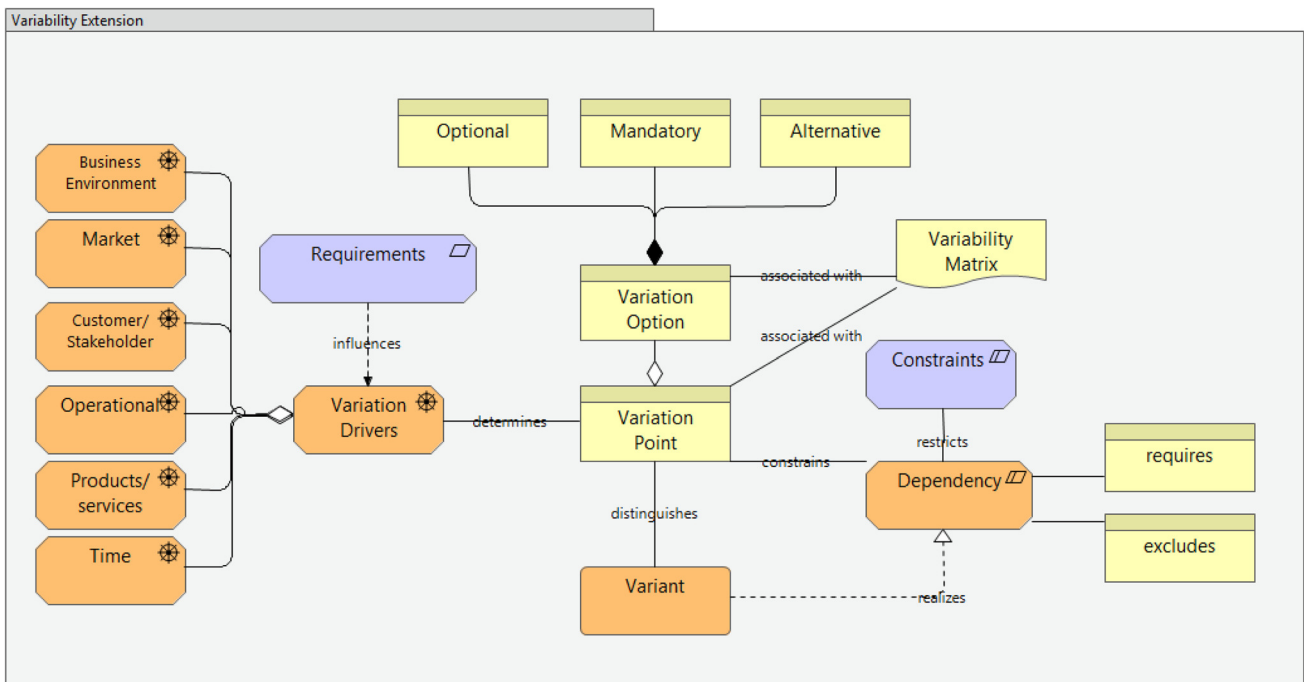


Fig. 2 Variability management concepts

dependencies, on their own, are restricted by constraints which may refer to business-related constraints, such as cost/budget during process design, or IT-related constraint, such as integration with vendor’s software systems. Given the constraints, the dependencies either may imply selection of a particular variant (require) or limit the possible options (exclude). A variation point can be resolved by many possible variants and a variant can resolve multiple variation points, so a many-to-many relation exists between variation point and variants.

5.2 Enterprise Architecture for Electronic Invoicing

We used the extended metamodel to design architecture models specific to the case study. We discuss two of the designed models to illustrate the use of the extended metamodel.

The first model is an L3 process model for *Invoice customer*, shown in Fig. 3. It is designed by a business process analyst together with business architects and business process owners. Given the detailed analysis of the fiscal regulations and the variability matrix for country-specific requirements (Table 2), we located branching points in the generic process of Invoice Customer. The model contains two variants of L4 activity *Transmit Billing*

data to customer. We decided to use a single-model approach in this case, since the variants are syntactically similar and have a significant part of the process in common.

The second model, the Function Allocation Diagram, shows the linkage between the variant of the L3 process, requirements, dependencies and supporting technologies in the allocation countries. It is designed by an IT architect together with business architects and business process owners. Because of the limited space, the diagram displays technology solutions and constraints only for two countries (see Fig. 4). The variants are represented using stereotypes in angle brackets, where “Opt_variant” and “Alt_variant” indicate an optional and alternative variant, respectively.

6 Evaluation

During the design process we involved primary stakeholders for iterative feedback and improvement of the solution. For the final evaluation, we decided to use the *analytical method* (Hevner et al. 2004) and explore the applicability/fit of the artifact into the EA of Philips with architecture designers who were not involved in the design process.

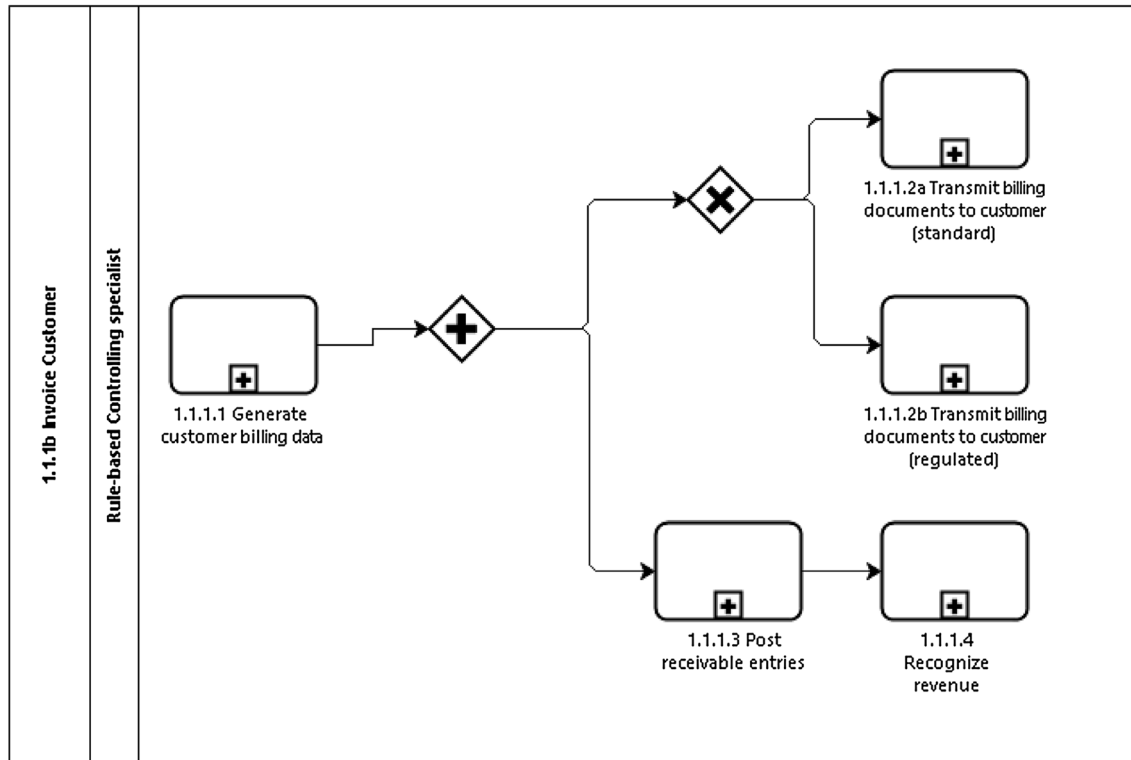


Fig. 3 Invoice customer process model

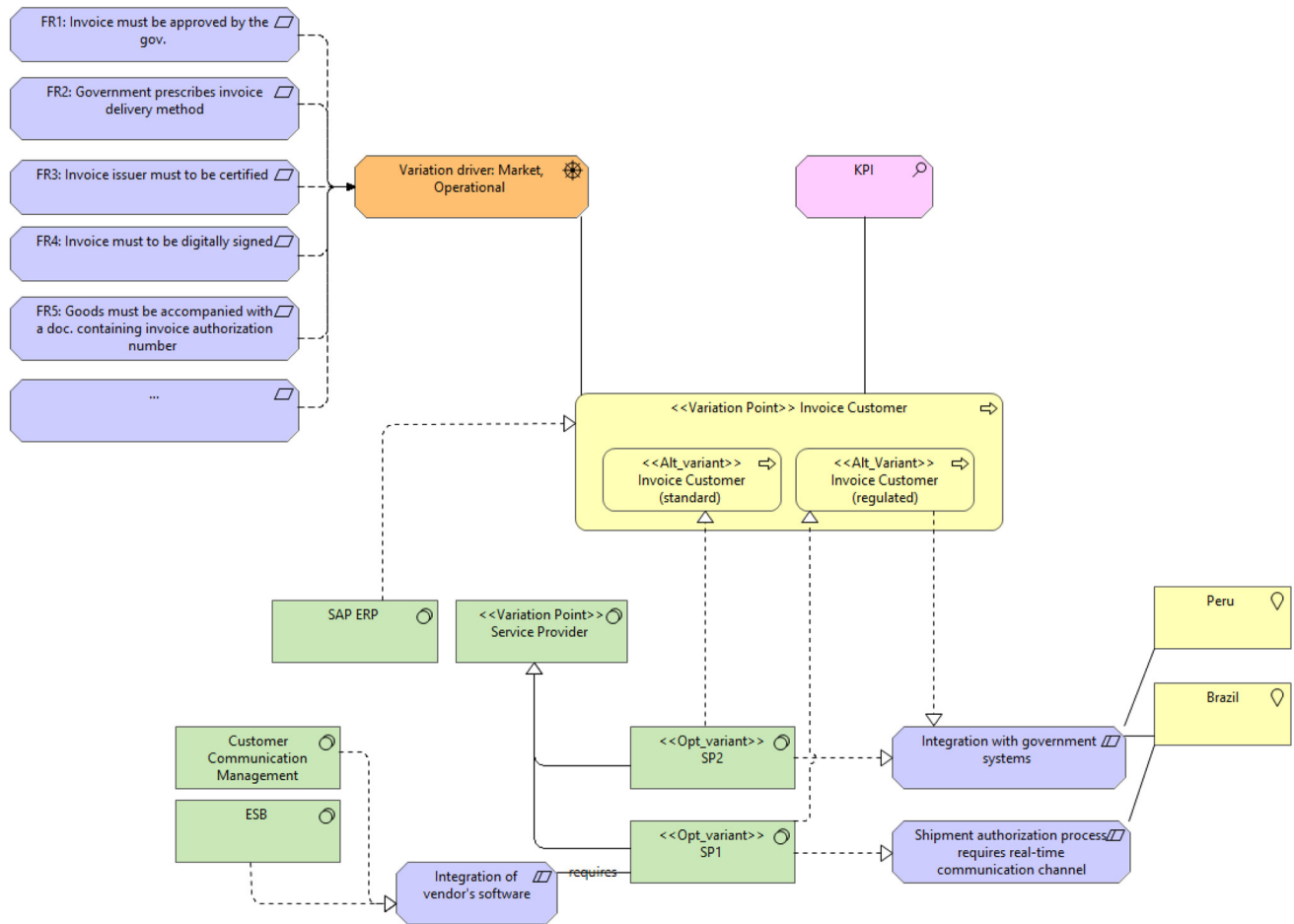


Fig. 4 Function allocation diagram

6.1 Evaluation Design and Results

The problem tackled in the research is improving the way variability is represented in EA. The developed EA meta-model extension is intended to be used and utilized by the stakeholders who are involved in EA design. To address the utility of the artifact, we were interested in the stakeholders' opinion about the consistency of documentation, understandability and applicability of the solution in the current setting (modeling tools and notations) of Philips.

The evaluation approach was conducted by performing interviews with three experts from Philips to discuss and evaluate the feasibility and applicability of the proposed solution. Other experts had been already heavily involved in the solution design process or were not available. The selected interviewees were a transformation architect (for business), a business architect (domain Order-to-Cash), and an enterprise architect, who represent key stakeholders in developing and guiding the design of the EA at Philips. From these three experts, the transformation architect had been lightly involved during the design process of the

solution. We had contacted the business architect and the enterprise architect mostly during the initial stages of the research to understand the approach and modeling standards used by Philips.

The evaluation session consisted of two parts. First, the solution and the case study were demonstrated. Next, an open-question interview session took place to receive feedback from experts on the quality and feasibility of the solution. The interviews were semi-structured. We chose this type of interview setup to be able to ask more detailed questions if necessary and to allow the interviewed stakeholders to elaborate more on a topic if needed. We asked open questions, listed in the appendix, about the consistency of the solution with its stakeholders and its intended purpose, its feasibility, and the expected usefulness of the solution. The questions are based on the validation aspects for software architecture documentation proposed by Clements et al. (2003). That source was selected because of its relevance to our research topic; the focus is on assessing both functional (if the solution meets intended purpose)

and quality aspects (consistent, understandable, compliant with company principles and usable by the stakeholders).

The overall evaluation of the proposed solution by the experts was positive; summaries can be found in the appendix. All of them emphasized the high relevance of the problem addressed in the research. They were also satisfied with the consistency of the solution and with its feasibility in the current EA landscape. All of them agreed the solution is applicable to the domain of Order-To-Cash, but a concern was the applicability of the solution to more complex cases of variability, for instance due to various customer demands.

6.2 Validity

Using a case study methodology in the design process creates an inherent threat to validity (Runeson and Höst 2009). We evaluate the *internal* and *external* validity aspects of the case.

The *internal* validity aspect concerns the causal relations investigated during the case study. It is important to be aware of factors influencing the design process. The threat to internal validity of our solution was addressed by spending a considerable amount of time getting familiar with the company environment and the tools and standards they use during the design. The considered factors include the overall architecture design principles and guidelines, different modeling techniques used for creating views in the EA, and perspectives of different stakeholders on the same processes. In addition to that, we tried to consider the fact that the company plans to switch to another modeling tool and related to that, several components of EA are going to be removed or renamed. We therefore abstracted several concepts to make them tool and language-independent and focused on representing links between the abstract concepts, to avoid that the developed solution and elaborated case study models only fit a specific tool or modeling language.

External validity of an artifact is concerned with to what extent the findings can be generalized. Conducting a single case study which affects only certain levels of business architecture from a single domain makes it hard to validate the generalizability of the solution to other cases. We reduced the threat of external validity by building our solution on established theories and techniques from the existing knowledge base on variability management in software engineering. Moreover, the case has multiple units of analysis, i.e. the different countries in Latin America. The case focuses on the domain of financial processes where variability is caused by different fiscal regulations. The solution applies to similar financial processes in other areas of the world, i.e., financial processes that have to adhere to government regulations, for instance

in countries in the European Union. As mentioned by the experts, applicability of the solution to other domains, in which variability is customer-driven and more complex, remains unclear. However, we do expect that the solution is applicable to other domains in which variability is caused by different regulations, e.g. healthcare.

The interviews with the experts also showed that handling variability in EA is especially important for them, because at the current EA maturity level almost all EA artifacts have been *defined*. The next step to a higher maturity level is to become able to *manage* the design process across all domains through a more standardized and consistent methodology. According to their feedback, the proposed solution gives a solid basis for adopting and standardizing it in the current architecture design.

6.3 Discussion

The formal evaluation involved stakeholders that guide the design of the EA models, i.e., the architects. In addition, the solution design has been aligned with a business stakeholder: the business process owner in Finance who was involved in collecting requirements and defining the process. By representing variability explicitly in EA models, we expect the complexity of EA models to become more manageable and maintenance efforts to be reduced. Assessing the impact of the approach on decision making is, however, difficult. It would require involvement of other stakeholders like the CIO (Lindström et al. 2006) as well as a clear attribution of the benefits of representing variability explicitly, e.g. saving costs by eliminating redundancies in the IT landscape or improving quality by explicitly showing which proven IT solutions are used to realize different variation drivers.

7 Conclusion

Though variability has been studied extensively in software engineering, managing variability for enterprise architectures has not been studied so far. In this paper, we have explored an initial solution for representing variability in EA by performing a case study of electronic invoicing at a multi-national organization. The solution design is an extended enterprise architecture metamodel based on ArchiMate. For the extension, we have added new elements covering concepts like variation driver, constraint, dependency and process variant. Variation points and their resolution options as well as dependency options are embedded (using stereotypes, layering, or branching points) within the models of specific architecture layers. The solution design helps the organization to represent

variability of different artifacts in their enterprise architecture and in the long run may help to reduce costs.

We expect that the solution can be applied to other financial processes, since financial processes such as electronic invoicing are fairly standardized at the high level, but have a lot of variations at a more detailed level. Also other domains in which variability is caused by regulations, e.g. healthcare, can benefit from the proposed solution. For domains in which more flexibility is allowed during the process design, e.g. in service-oriented markets that target customers with various demands, a more flexible solution to variability management may be needed.

In future work, we plan to apply the design to case studies in which variability is due to customer demand. Next, this paper studies the design phases of variability management in enterprise architecture, including the identification, documentation and modeling. Other future work can address the instantiation and implementation phases, where run-time execution of specific variants with supporting information systems can be explored in more details. Finally, another interesting topic is how to resolve conflicts caused by variability constraints at different levels.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Atkinson C, Gerbig R, Fritzsche M (2015) A multi-level approach to modeling language extension in the enterprise systems domain. *Inf Syst* 54:289–307
- Ayora C, Torres V, Weber B, Reichert M, Pelechano V (2015) VIVACE: a framework for the systematic evaluation of variability support in process-aware information systems. *Inf Softw Technol* 57:248–276
- Azevedo CL, Iacob M-E, Almeida JP, van Sinderen M, Pires LF, Guizzardi G (2015) Modeling resources and capabilities in enterprise architecture: a well-founded ontology-based proposal for ArchiMate. *Inf Syst* 54:235–262
- Bērziša S, Bravos G, Gonzalez TC, Czubayko U, España S, Grabis J et al (2016) Capability driven development: an approach to designing digital enterprises. *Bus Inf Syst Eng* 57(1):15–25
- Chen L, Babar M (2011) A systematic review of evaluation of variability management approaches in software product lines. *Inf Softw Technol* 53(4):344–362
- Clements P, Bachmann F, Bass L (2003) Documenting software architectures: views and beyond. Addison-Wesley, London
- Czarnecki K, Grünbacher P, Rabiser R, Schmid K, Wąsowski A (2012) Cool features and tough decisions: a comparison of variability modeling approaches. In: Sixth international workshop on variability modeling of software-intensive systems. ACM, pp 173–182
- de Kinderen S, Gaaloul K, Proper HA (2014) Bridging value modelling to ArchiMate via transaction modelling. *Softw Syst Model* 13(3):1043–1057
- Economist (2014) Electronic arm-twisting. <http://www.economist.com/news/finance-and-economics/21602274-reduce-tax-fraud-governments-encourage-automated-accounts-electronic>. Accessed 24 Nov 2017
- Engelsman W, Quartel D, Jonkers H, van Sinderen M (2011) Extending enterprise architecture modelling with business goals and requirements. *Enterp Inf Syst* 5(1):9–36
- Erol O, Sauser BJ, Mansouri M (2010) A framework for investigation into extended enterprise resilience. *Enterp Inf Syst* 4(2):111–136
- Galster M, Avgeriou P (2015) An industrial case study on variability handling in large enterprise software systems. *Inf Softw Technol* 60:16–31
- Hallerbach A, Bauer T, Reichert M (2010) Capturing variability in business process models: the Provop approach. *J Softw Maint Evol* 22(6–7):519–546
- Hevner A, March S, Park J, Ram S (2004) Design science in information systems research. *MIS Q* 28(1):75–105
- Iacob ME, Meertens LO, Jonkers H, Quartel D, Nieuwenhuis L, van Sinderen M (2014) From enterprise architecture to business models and back. *Softw Syst Model* 13(3):1059–1083
- Koch B (2014) E-invoicing/e-billing – key stakeholders as game changers. Billentis, Wil
- Korman M, Lagerström R, Ekstedt M (2016) Modeling enterprise authorization: a unified metamodel and initial validation. *Complex Syst Inform Model Q* 7:1–24
- La Rosa M, van der Aalst W, Dumas M, Hofstede A (2009) Questionnaire-based variability modeling for system configuration. *Softw Syst Model* 8(2):251–274
- Lagerström R, Johnson P, Höök D (2010) Architecture analysis of enterprise systems modifiability – models, analysis, and validation. *J Syst Softw* 83(8):1387–1403
- Lankhorst M et al (2013) Enterprise architecture at work, 3rd edn. Springer, Heidelberg
- Lindström Å, Johnson P, Johansson E, Ekstedt M, Simonsson M (2006) A survey on CIO concerns – do enterprise architecture frameworks support them? *Inf Syst Front* 8(2):81–90
- Manzur L, Mario Ulloa J, Sanchez M, Villalobos J (2015) xArchiMate: enterprise architecture simulation, experimentation and analysis. *Simulation* 91(3):276–301
- Mikaelian T, Nightingale DJ, Rhodes DH, Hastings DE (2011) Real options in enterprise architecture: a holistic mapping of mechanisms and types for uncertainty management. *IEEE Trans Eng Manag* 58(3):457–470
- Milani F, Dumas M, Ahmed N, Matulevičius R (2016) Modelling families of business process variants: a decomposition driven method. *Inf Syst* 56:55–72
- Object Management Group (2011) Business process modeling notation (BPMN). Object Management Group. <http://www.bpmn.org/>. Accessed 28 Nov 2017
- Park J, Moon M, Yeom K (2011) Variability modeling to develop flexible service-oriented applications. *J Syst Sci Syst Eng* 20(2):193–216
- Pohl K, Böckle G, Linden F (2005) Software product line engineering: foundations, principles and techniques. Springer, Heidelberg
- Rosemann M, van der Aalst W (2007) A configurable reference modelling language. *Inf Syst* 23(1):1–23
- Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. *Empir Softw Eng* 14(2):131–164
- Schmid K, Isabel J (2004) A customizable approach to full lifecycle variability management. *Sci Comput Program* 53(3):259–284
- Sinnema M, Deelstra S (2007) Classifying variability modeling techniques. *Inf Softw Technol* 49:717–739

- Sinnema M, Deelstra S, Hoekstra P (2006) The COVAMOF derivation process. ICSR'06. Springer, Heidelberg, pp 101–114
- Svahnberg M, Gulp J, Bosch J (2005) A taxonomy of variability realization techniques: research articles. *Softw Pract Exp* 35(8):705–754
- The Open Group (2011) TOGAF version 9.1. <http://pubs.opengroup.org/architecture/togaf9-doc/arch/>. Accessed 24 Nov 2017
- The Open Group (2013) ArchiMate 2.1 specification. <http://pubs.opengroup.org/architecture/archimate2-doc/>. Accessed 24 Nov 2017
- Wilkinson M (2006) Designing an 'adaptive' enterprise architecture. *BT Technol J* 24(4):81–92
- Winter R, Fischer R (2007) Essential layers, artifacts, and dependencies of enterprise architecture. *J Enterp Architect* 3(2):7–18
- Winter K, Buckl S, Matthes F, Schweda CM (2010) Investigating the state-of-the-art in enterprise architecture management methods in literature and practice. In: Conference: the 5th mediterranean conference on information systems, MCIS 2010, Tel-Aviv-Yaffo academic college, Tel Aviv, Israel, September 12–14
- Zachman JA (1987) A framework for information systems architecture. *IBM Syst J* 26(3):276–292

Representing Variability in Enterprise Architecture

A Case Study

Nino Rurua, Rik Eshuis, Maryam Razavian

Business & Information Systems Engineering 61(2) (2019)

Appendix (available online via <http://link.springer.com>)

Interview questions

The questions used during the evaluation sessions are given below:

1. Is the approach consistent with the stakeholder community who's going to use it (enterprise, business, IT architects, business process owners and experts)?
2. Is the approach consistent itself?
 - a. Are the mappings between different steps, elements of the approach provided?
 - b. Are there any ambiguities, redundancy, limitations?
3. Does the documentation adhere to the standards and templates that it claims to follow?
4. Is the approach consistent with the purpose it describes?
5. Will you consider using the proposed variability extension elements in the architecture design?

Evaluation Results

A short summary of the evaluation feedback from the interviewed stakeholders is provided below:

Expert 1: Transformation Architect

1. The approach is consistent with the stakeholder needs: the problem reflects the company's current needs and is highly relevant to the issues faced in the design of enterprise architecture. The proposed approach is in line with architecture methodology used at Philips.
2. The concepts about variability are clear, the steps for variation identification are easy to follow, and there are no redundancies. The reasoning behind the selection of variability modeling approaches is understandable but not clearly explained.
3. The documentation complies with the Philips modeling standards and is applicable with the tools used for documenting enterprise architecture.
4. The purpose was to understand how to analyze and document variability, and what are the ways to do it, and the proposed approach provides it. However, it gives one example of how it can be applied in practice.
5. The suggested variability concepts are very useful. They give the content and substance to the issues the company faces which was not clearly defined before.

Expert 2: Business Architect (Order-to-Cash)

1. The approach is consistent with the stakeholder community and is a big step further to solve problems the company has in the design, particularly unstructured and dispersed approaches how to manage variations. It shows how the company can adapt its current classification of the processes in the architecture which can be the next stage of the current architecture maturity.

2. The structure and the options are clear, and also the process how to follow variability management from its identification to modeling. It would be helpful to see more generalized overview of the pros and cons of modeling approaches which cover other cases of variability as well.
3. The documentation of the models is consistent and follows the company's modeling standards. It would be easier for the stakeholders to read the proposed process of variability management if modeled using BPMN notation since the same language is used for business process modeling.
4. The proposed approach is consistent with the purpose and how it deals with the e-invoicing case. But it will be interesting to have more understanding how the same approach can be applied to more customer/user-specific variability issues which can be more complex in nature than finance processes.
5. The proposed solution guides the designers to make decisions about variations and which steps to take further to embed it in the current modeling techniques. Especially, having the classification of the variation drivers and constraints that have to be considered during the design of business architecture. Classified causes of variation enables more standardized approach and reduces redundancies in the modeling which is common across domains currently. The aspect which would help the architects more and could be the next step is to define the ways how to execute the identified variants.

Expert 3: Enterprise Architect

1. The approach is consistent with the stakeholders who are going to use it. The metamodel gives a formal way of describing the issue and that should help to easily apply it to the work. But the question is if all the stakeholders are able to recognize and read the metamodel. For example, during communication giving more concrete examples of variation drivers and textual explanation could help the business process experts or owners (who are less familiar with ArchiMate language) to understand the purpose of the elements.
2. The overall approach is useful and consistent. Having variation drivers and variation points seems redundant, as they seem to capture the same context. Instead of variation points, it would also be possible to have variation sub-drivers to capture more specific variability cause/location.
3. The approach adheres to the modeling language (ArchiMate) and the viewpoints used in the enterprise architecture. There are some differences with the current labels of the objects, but that is due to the recent switch to a new tool.
4. The approach for enterprise architecture is consistent overall but the application layer is not covered as extensively as the other layers. However, variability affects application architecture in a different way (configuration templates) and is not that important for this approach.
5. The proposed approach is useful to have a formal way of analyzing and describing variation issues for business processes and technology component support, but the application layer support is weak, so can be left out.