Industry Solution adaptability

An integrated solution model and an integrated analysis and engineering method addressing change in large and complex enterprises

Wiel Bruls[*,a,b], Edward Giesen[a], Marc Lankhorst[c], Robert Winter[b], Hans Slaets[a]

[a] IBM Global Business Services, Amsterdam, The Netherlands
[b] University of St. Gallen, Institute of Information Management, St. Gallen, Switzerland
[c] BiZZdesign, Enschede, The Netherlands

Abstract. *The spectrum of changes that enterprises need to deal with varies from simple continuous adjustments of the product portfolio in response to evolving customer preferences, to complete overhauls of the business and operating model in response to disruptive trends. Many research fields and practitioner disciplines have produced analysis and engineering approaches that can help enterprises to assess and prepare for the impact of changes from this spectrum. However, they have partial scopes and consequently limited integration. By selecting, slightly extending and integrating existing approaches, this paper introduces a 'simple enough' integrated solution model and a 'simple enough' integrated analysis and engineering method that covers the full spectrum of changes.*
Our focus is the large, complex enterprise that operates in a specific industry and performs information processing at scale. The research is intended to provide methodical support to practitioners with a responsibility for shaping solutions. Our proposal is the result of initial experiences in practice that instilled the research theme, application in a large-scale industry project, focused collaborative research that joined researchers and academia, and ongoing applications and experiences in practice. The solution model and the analysis and engineering method that we propose support three types of adaptability: a) foundational adaptability produces full new business model and operating model parts, b) transitional adaptability extends the current business model and operating model and adds additional configurability, and c) routine adaptability is managed within the configurability of individual operating model components that need to be designed with sufficient bandwidth. A business configuration center is proposed as a key constituent that manages the differences in underlying technology, and that allows to perform integrated, technology agnostic administration of an industry solution.

Keywords. Adaptability • Industry solutions • Configurability • Business model • Operating model • Variability modelling • Feature modelling • Business Configuration

Communicated by Agnes Koschmider. Received 2019-05-17. Accepted after 4 revisions on 2020-06-29.

1 Introduction

Large enterprises are complex adaptive systems that are continuously subject to change pressures that require them to adapt their business model and their operating model. In the past the enterprise IT solution landscape has evolved considerably, however in a quite linear fashion. First focused on internal production processes, with support for internal administrative processes added in a next step, and then opening IT systems to support customers and partners. Enterprises had few options

* Corresponding author.
E-mail. wiel.bruls@unisg.com

but to follow this sequence of changes, and change could be handled, therefore, as a single projected change path towards a future state.

Today, however, the number of options for change that enterprises can choose from has increased drastically by coinciding business and technology trends. Examples of business trends include digitization of products and services (Bharadwaj et al. 2013; Klinger and Beckner 2012), digital ecosystems of cooperating partners such as at harbors and airports (Bruls et al. 2010), application platforms for smartphones, gaming, and commercial software packages (Basolea et al. 2015; Darking 2007; Jansen et al. 2013; Wareham et al. 2014), partner networks for delivering integrated products and services (Chau et al. 2011; Demirkan and Spohrer 2016; Devadoss 2014), online interacting communities (Marshall et al. 2014), and sustainable manufacturing (Pichagonakesit et al. 2019). Examples of technology trends include ubiquitous sharing of online information (Fung and Hung 2013) and connectedness of devices (Botta et al. 2016; Marshall et al. 2014; Whitmore et al. 2013), cognitive, analytic, and collaborative technologies (Baird and Parasnis 2011; Kelly and Hamm 2013; Lim et al. 2013), augmented reality, digital twins, geospatial integration (Blaga and Tamas 2018; Eldrandaly et al. 2019; Revetria et al. 2019), data science and artificial intelligence (Aparicio et al. 2019), and cloud-based delivery of solutions (Cusumano 2010; Wang et al. 2010).

Changes from these trends combine with the more regular adjustments such as updates to a product portfolio to meet evolving customer preferences. Together these options create a highly volatile wicked mix (Girod and Whittington 2019; Tanriverdi et al. 2010) with very different impacts on the enterprise landscape (Giesen et al. 2007; Girod and Whittington 2019; Tanriverdi et al. 2010). *Foundational* changes due to disruptive trends impact the overall structure of the operating model and business model, *transitional* changes due to new technologies, products and production methods require extensions of the operating model and business model, and *routine* changes due to evolving customer preferences and product updates require adjustments of existing parts of the business and operating model.

Motivation and problem statement

To prevent being stuck in a swamp of incompatible ad hoc extensions, enterprises need an approach that allows them to prepare for the impact of this broad spectrum across a horizon of years. This implies both analyzing and engineering the required types of adaptability across the Business to IT (B2IT) stack that the enterprise has deployed. If a change then needs to be implemented it is foreseen, and side effects can be limited.

Many different research fields have addressed change. Examples include business model innovation analysis (Giesen et al. 2007), architectural modifiability analysis at both enterprise (Rurua et al. 2017) and software component level (Bengtsson et al. 2004), and solution variability and configurability analysis (Galster and Avgeriou 2013). However, all of these are standalone research fields with partial scopes and limited integration. On the practitioner side, methods such as TOGAF and MSP support change (MSP 2021; TOGAF 2021) as well. It is, however, a 'linear' concept: the transition from one state to a next state. Integration exists such as between architecture methods and engineering methods through shared modelling entities like processes and functions (Lankhorst 2017) that model commonality. However, there is no alignment of adaptability modelling across them that supports the wide spectrum of changes that we seek to address in this paper. And also here, the number of analysis and engineering methods that are required to cover the full spectrum is considerable. They include strategic analysis, enterprise architecture structuring, software architecture of applications, and software engineering of individual components that aims for configurability.

Without integration into a method that drives for simplification, analysis of adaptability is cumbersome and integration of different approaches requires translations between models and creates complex dependencies. Some expert users may be able to grasp them, but the results would be difficult to follow certainly for business stakeholders. The latter ones have a key stake in understanding if future technology changes and routine changes in a product portfolio can be supported by their operating model - without incurring major costs and large delays in time to market.

Research question

The research question we pursue is twofold (Thuan et al. 2019):

- RQ 1: Which 'simple' enough *industry solution model* can incorporate the spectrum of changes from routine, through transitional to foundational?
- RQ 2: How can the required adaptability be designed through a 'simple' enough *analysis and engineering method*?

Basic Principles of our Approach

We build on the extant literature of existing enterprise level methods for strategic, architecture and engineering analysis, with a limited number of extensions. As *solution model* we will propose a three layered B2IT stack as a base, that describes how the business is organized, how it is run, and what are the technical solutions. As *analysis and engineering method* we will propose a method that consists of separate strategy, architecture and engineering phases, that assess impact and architect and design the required adaptability. The basic design principle that we apply is that of designed adaptability (Galster and Avgeriou 2013): 1) foreseeing a spectrum of changes across a horizon of years and assessing their impact on the B2IT stack, 2) structuring the design of the B2IT stack in such way that it can be adapted when the change occurs.

Key contribution and intended audience of our research

We consider the solution model along with the integrated analysis and engineering method as the key contribution of our paper. As recipients of our research, we address practitioners with a responsibility for shaping enterprise strategy and structure such as enterprise architects and engineers and innovation managers that act in transition and innovation management roles (De Haes and Van Grembergen 2009). And we address the research communities from research fields that are interested in an integrated approach to designed adaptability.

Structure of the paper

In the paper sections that follow, we first discuss the research method, and derive requirements. The main body of the paper starts with a high-level overview of the solution model, and the analysis and engineering method. It is followed by detailed sections for both. In the ending section we evaluate the approach, and discuss future developments. Three supporting appendices are included: available methods and how we use them, examples from the practice of the authors that illustrate drivers for the approach, and an application of the approach for a large institution in a specific government sector.

2 Research Method

As the outcomes of our research are purposefully designed artefacts that can be instantiated, we follow the Design Science Research (DSR) approach (Hevner et al. 2004; March and Smith 1995; Simon 1996). For the research process, we follow the detailed steps proposed by Peffers et al. (2007). See Fig. 1 for an overview. The approach that we describe is the result of initial experiences in practice that instilled the research theme, application in a large-scale industry project, focused collaborative research that joined researchers and academia, and ongoing applications and experiences in practice. The different iterations resulted in incremental development of the analysis and engineering method. First producing the core architectural componentization based upon trend impact assessment, then support for operational differences across implementations and integrated configuration to cope with multiple disjunct metadata of the various products and technologies in the solution, next development of a feature language for routine configurations within predefined bandwidths, and in the last iteration integration with

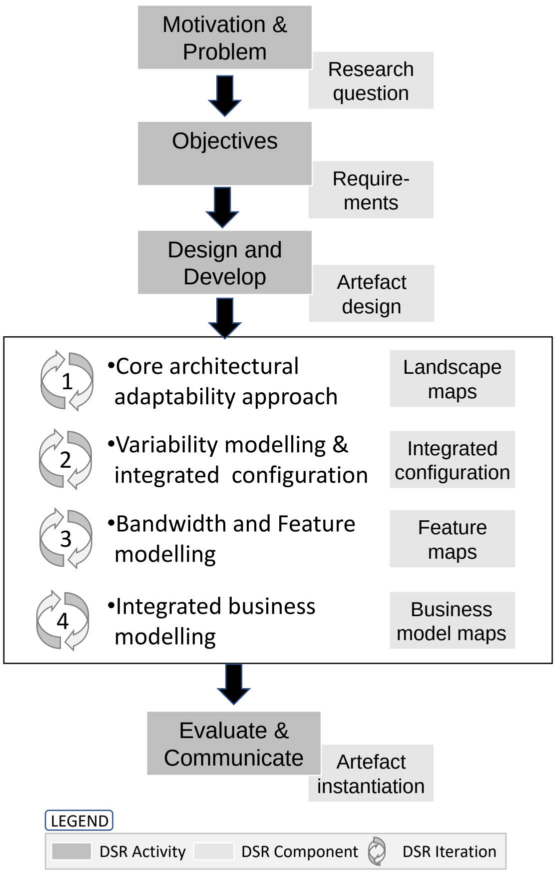

Figure 1: Research process

the business model. The first three iterations were performed in the context of engagements by practitioners. The final largest iteration was performed as a separate research project that included experienced business strategy and enterprise architecture practitioners with a broad research background as well, enriched with academic researchers.

Evaluation and communication of the proposed model and method has been performed in every phase of the research. The initial model and method were used for example to guide choices in the architecture review board overseeing development of an industry solution for the customs sector.

3 Scope and requirements

In this section we define the scope that we intend to address, and the requirements that our artefacts must meet.

3.1 Scope

Our scope in this paper is the large enterprise with a large and complex set of enterprise information systems. This includes the pre-Internet enterprise focused on production of physical goods and services such as consumer goods, travel and transport, insurance and financial services, et cetera. Those that have been born since then and focus fully on online services such as online search, community sites, digital contents, et cetera. And the hybrids that combine both in either product or in business operations such as commerce websites with physical/digital products, marketplaces, et cetera.

Information landscapes in these enterprises are layered and populated with information systems of very different types and assembled with many technologies. They may include custom developed websites for use by consumers, customers or partners, packaged applications for managing sales and customer relationship, integration infrastructures that connect systems, production control systems that drive transaction processing of digital services or manufacturing of physical products.

What we exclude from our research are new types of applications with local autonomous intelligence of much smaller scale. This includes cyber-physical systems (for example self-driving cars, swarms of drones) in which independent entities interact with a complex heterogeneous environment, communicate with similar entities and develop joint strategies (Gerostathopoulos et al. 2017). Embedded IT applications (e.g smart TVs) that can control increasingly reactive devices and need to support a variety of consumer preferences (Capilla et al. 2014). Physical or virtual assistants that interact with humans and need to be able to interpret behavior and language, reason and control movements, et cetera.

Solutions for these systems are characterized by software that implements local intelligence that can interact and coordinate with external entities, with limited central coordination and design, and with software solutions that are complex, but of limited scale. While some of the proposed analysis methods can be applied to these systems, they generally have different adaptability needs.

3.2 Requirements

In this section we derive key requirements[1] that guide the development of the analysis and engineering method and the solution model. Tab. 1 includes for each requirement a short title, a description of the requirement and the artifact characteristics that support them. The latter are described in follow-on sections, but a cross linkage is already included here to allow lineage and traceability. Origin and rationale of the requirements can be understood as follows:

Requirement 1 addresses the scope of applicability and within this scope the differentiation. The large-scale solution landscapes in the enterprises that we address (see Sect. 3.1), depend to a considerable degree on the specific industry (Flaxer et al. 2005; Harishankar and Daley 2011; Pohle et al. 2005). For example, manufacturing of physical products, delivery of financial services, delivery of telecommunication services, provisioning of travel and transport services, all require very different solutions for internal production (from physical production line control to processing of administrative data in transactions) and external interactions (from catalogue-based buying through intermediaries to relationship-based recommendations). Therefore, the analysis and engineering models we introduce should support industry specific modelling.

Requirement 2 – 4 address the basic premises that we have adopted for developing the artefacts: 'simple enough' and built from existing approaches. These premises tie back to the complex mix of changes that enterprises are subject to (Giesen et al. 2007; Girod and Whittington 2019; Tanriverdi et al. 2010) that we aspire to provide solutions for with the minimum amount of complexity and by leveraging existing well-known methods.

Requirement 5 addresses the need to anticipate foundational changes and contain their impact. The analysis and engineering methods that we introduce should aim to structure the operating model (Campbell et al. 2017) in such way that the impact of a trend can be contained with limited side effects (see the first Example from practice in Appendix B). Consider, for example, an enterprise that decides it wants to consider future monitoring of the products it ships using signals produced by sensors integrated into its products. Then our analysis that produces the architectural partition in the operating model that will contain components that receive and process these monitoring signals, should isolate it from other areas of the operating model.

Requirement 6 addresses the need to anticipate routine changes and match solutions to the expected spectrum of changes. The required investments to produce solutions in the large-scale enterprise are considerable. Once a decision has been taken on for example the key functionalities and the extent to which these should be adaptable (Winter 2011), the effort to transition out of these solutions can be very large, resulting in barriers to change. To prevent crashing into these barriers, this raises the need for an approach that can match the extent to which a solution can be adapted to the routine variations that are expected (see the second Example from practice in Appendix B). Therefore, alignment of adaptability for routine changes in business model and operating model is a key characteristic. Consider, for example, a financial services company that decides to extend its coverage of insurance services to new customer segments with new eligibility rules, that require collection of additional context details from the customer. The scope of adaptability of the business functions in the operating model that collects these details should have been designed in

[1] As DSR research is a discovery process that searches for artifact characteristics in a broader space, requirements have a different flavor than in software engineering. They do not intend to provide a full functional specification, but are more aimed at scoping and high–level delineation (Maedche et al. 2019).

No	*Requirement Short Title*	*Requirement Description*	*Artefact Characteristics*	*Described in*
1	Large Enterprise Scope, Industry specific	Support adaptability of the large enterprise with a large and complex set of enterprise information systems, in a specific industry	Usage of enterprise level analysis and engineering methods for adaptability that produce industry specific models	Sect. 4
2	'Simple Enough' Solution model	Support a 'simple enough' solution model and 'simple enough' adaptability techniques that can incorporate with minimum complexity the change spectrum that the large enterprise needs to cope with	Support for routine reconfigurations, transitional extensions and foundational overhauls of a three-layered solution model using basic adaptability providing techniques	Sect. 5
3	'Simple Enough' Analysis	Support a 'simple enough' method, that can perform with minimum complexity the adaptability analysis and engineering required to resolve the impact of the spectrum of changes on the solution structure	A three-phased analysis and engineering method, that supports integrated, transparent and understandable modelling of adaptability at the various levels of change	Sect. 6
4	Build from existing approaches	Integrate existing approaches with limited extensions	Harvest from existing enterprise level methods and integrate adaptability modelling into these	Sect. 6
5	Trend impact containment	The impact of trends on the industry solution should have limited side effects (see 1st example in Appendix B)	Architectural partitioning and componentization using landscape views with dimensions selected that aim for containment of changes	Sect. 6.2.1
6	Optimum bandwidth	Modelling of routine adaptability should appreciate that once solutions with certain adaptability range have been developed these are hard to extend beyond the original scope (see 2nd example in Appendix B)	Optimum bandwidth modelling	Sect. 6.2.3
7	'Simple enough' configurability	Support a 'simple enough' configuration language (intuitive and easy to understand for business users with limited knowledge of technology) that can integrate multiple types of metadata (see 3rd example in Appendix B)	Integrated feature models managed from a business configuration center that envelopes the metadata from existing technology stacks	Sect. 6.3
8	Technology agnostic	Support analysis and solution techniques that are technology agnostic	Generic adaptability providing techniques that include architectural structuring and partitioning, and generic elements with configurable features, with no prerequisites in technology	Sect. 6.2, Sect. 6.3

Table 1: Key requirements, artifact characteristics that support them, and reference to relevant document section.

such a way that the required change can be easily configured.

Requirement 7 – 8 address the need to support technology independent and easily understandable configuration. Enterprises are used to a large spectrum of different technologies and solutions and the different ways that these can be customized and configured (Lankhorst 2012, 2017). Commercial off-the-shelf application packages acquired from vendors, bespoke developed applications, middleware technologies, all have their own approach to configuration. With the increasing dynamic nature and pace of technology developments, the approach that we introduce should, therefore, be able to work with different technology paradigms and different adaptability implementing techniques. This requires a technology independent configuration mechanism that can be integrated on top of technology specific mechanisms. The configuration 'language' should be intuitive and easy to understand for business users, with limited knowledge of technology, and of limited complexity (see the third Example from practice in Appendix B).

4 Artefact Overview

This section introduces the artefacts that we deliver: Integrated solution model and Analysis and Engineering method (see Fig. 2). They are detailed in follow-on Sect. 5 and Sect. 6. The Integrated solution model (top part of Fig. 2) will be used by practitioners to model the change impact at three levels: the business model, operating model and technology model. The B2IT stack organized into these three layers reflects the core of what an enterprise does: producing (in an operating environment) for a purpose (the business model) supported by technology (the technology model). It is 'simple enough' for the modelling of impacts across the spectrum of changes. The business model allows to model both impact of routine changes to the business, as well as of completely new value propositions and disruptions of the business model. The operating model allows to model the impact on processes and functions with

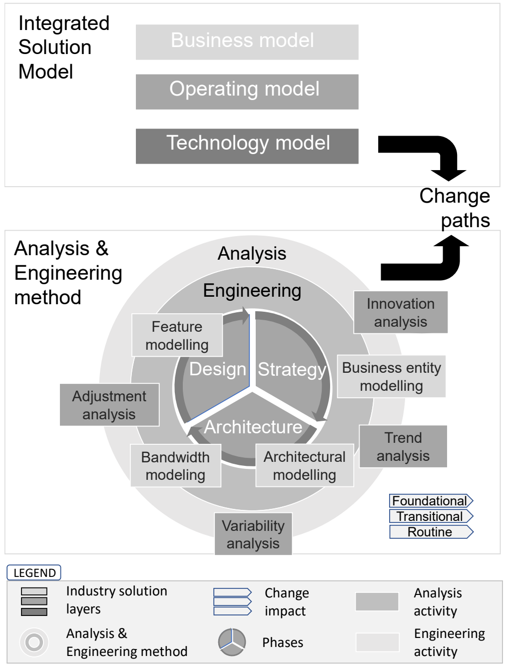

Figure 2: Artefact overview

full new partitions added, new generic elements or just configuration of existing elements. And the technology model supports these changes with new technology components, or extensions.

When comparing the three layers to an enterprise architecture stack as in ArchiMate (Lankhorst 2017), then they cover the strategy, business and application layer. As our analysis is technology agnostic we stop there. The business model layer comes from strategy research and is not directly represented in this way in EA methods but can be mapped (Meertens et al. 2012).

The Analysis and Engineering method (bottom part of Fig. 2) will be used by practitioners to perform adaptability analysis and engineering in three basic phases: Strategy, Architecture and Design (shown in the center right of Fig. 2). In all three phases, the spectrum of foundational, transitional and routine adaptability is considered. These phases are at the core of business IT alignment in methods such as strategic change, enterprise and software architecture, and business and IT service design (Harishankar and Daley 2011; Henderson and Venkatraman 1993; Lankhorst 2012, 2017). The approach is 'simple enough' for analysis and design of the results of the impact across the spectrum of changes. The analysis activities such as innovation, trend, variability and adjustment analysis, and design activities such as business entity, architectural, bandwidth and feature modelling, are specific for each phase and are explained in Sect. 6. The outcome of the impact assessment process is a full set of change paths (at the right in Fig. 2). The enterprise (or the provider of an industry solution) can prepare itself for these change paths by 'prewiring' into the design at the different layers the adaptability properties that are required to support the change. For example, identifying architectural partitions and components that may be required in the future while already now preparing current partitions and components to anticipate interactions. Or identifying the range of features that a future product will require and preparing components in the production infrastructure to support these. If, and when the enterprise then selects to implement the change, the enterprise solution model is prepared for it and the change can be implemented with limited effort.

5 Solution Model and Techniques

The three-layered solution model introduced in Fig. 2 is elaborated in more detail in Fig. 3. (For an overview of the extant literature, and how we build on that see Appendix A.1). The middle of Fig. 3 depicts the three layers[2] in the B2IT stack, with example data taken from the case of a Customs agency that is overseeing Import, Export and Transit of goods.

Layers are defined as follows. The business model is the fundamental organization of the business (Giesen et al. 2007; Osterwalder et al. 2005) that contains business model design entities such as the value proposition, products, customers, channels, processes, resources, etc. The operating model is the fundamental organization of the production environment (Campbell et al. 2017; Ross et al. 2006), that hosts the core operating model components (Sessions 2008) that consist of packages of business functionalities that support running the business (Campbell et al. 2017; Flaxer et al. 2005; Pohle et al. 2005; Ross et al. 2006). The technology model is the set of technology model components that support automation of business operations, with a mix of implementation technologies.

Constructs at the right in Fig. 3 reflect the commonality in the solution: business model elements, operating model components, and technology model components. They contain generic contents within certain boundaries. For example, operating model components will contain a generic set of business functionalities such as process models, activity models, information models, function models, etc. And technology model components will contain a set of generic capabilities at the technology layer, for example provided by an ERP-package that supports the financial account structure and processing, a process orchestration engine that supports specific sequencing of production processes, etc. Generic models will support a certain bandwidth of change that reflects the spread in the current business environment with a certain horizon into the future. In the example of a customs solution, the variations that we encountered in iteration 2 (see Sect. 2) in clearing goods at the border is such a bandwidth. They can be halted at entry until taxes have been paid, or for trusted traders they can be allowed to pass with tax levying independent of that. The bandwidth modelling that supports these variations may be performed by a provider of an industry solution with multiple customers with different requirements. Or it may be performed within an enterprise to model the possible spread that the current business environment may bring.

At the left of Fig. 3 the two techniques are depicted that support adaptability:

[2] Following Abraham et al. (2013), who apply the theory of hierarchical, multilevel systems to EA management, we identify the three structures in the B2IT stack as 'layers'.

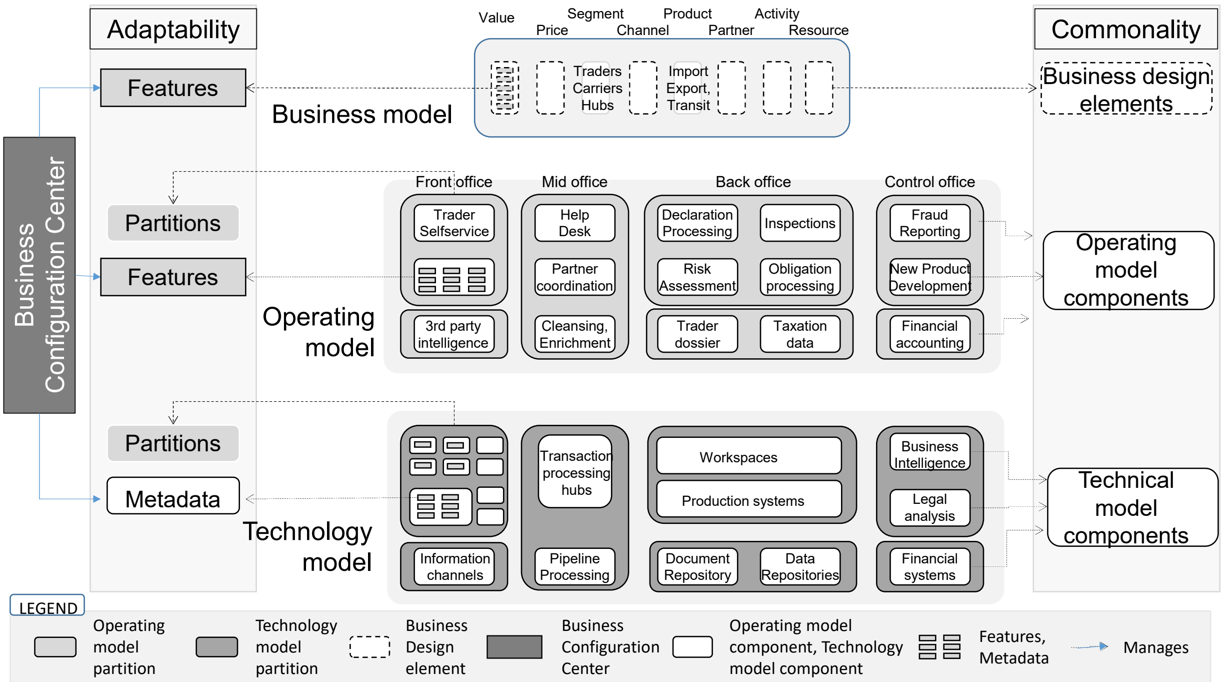

Figure 3: Integrated Solution model

- Partitioning and componentization to produce partitioned and componentized landscapes at the level of both the operating model as well as the technology model
- Feature/metadata modelling to allow configuring generic models within their defined bandwidths at all the three layers.

Partitioning and componentizing

These are well-known, proven and established architectural techniques that can isolate the impact of change. Partitioning segments the operating model landscape into a topology that creates proximity between those components that communicate most frequently (Harishankar and Daley 2011; Torre et al. 2013). Componentization creates operating model components as coherent well-bounded bundles of generic business functionality with bandwidths of predefined variability scope, that interact at interfaces only. Together these techniques can contain the impact of disruptive changes by enveloping impacts, reducing touch points between partitions and components and limiting side effects. An example would be a partition that is created to hold operating model components that support interactions with external customers, with different components supporting corporate and consumer customers.

Feature modelling

Feature modelling (Kang et al. 1990) is a specific implementation of metadata-based adjustments of generic domain models (Haugen 2013). Metadata based adaptation of generic models in business and software components is the established approach in 1) package technologies (Strong and Volkoff 2005), 2) model driven middleware-based development (Song et al. 2015; Wasilewski 2015), or 3) model driven assembly using APIs (Lankhorst 2012). The format of the metadata depends on the specific functionality and the implementation technology. For example, a Taxonomy table for an organizational structure in a package application, a Business Process Execution Language (BPEL) script for a process model in a process orchestration engine, a Mapping template for services adjustments in an API, etc.

Feature design leans on the Product Line Engineering (PLE) approach that originated in the mass customization manufacturing industry (Blecker and Friedrich 2006). PLE designs a product portfolio out of families of products with selectable features. Products are assembled in a factory by programmable machines that construct the product with the selected features. For example, in the automobile industry a family of compact cars is introduced with options that can be customized, such as different paint colors, different engine types, etc. PLE has been carried over to IT with the aim to produce "adaptive applications built as component-oriented system families with variability modeled explicitly as part of the family architecture" (Hallsteinsen et al. 2006; Pohl et al. 2005). Applied to IT, it has been extended to model variability of operational constructs such as Web services and processes (Galster and Avgeriou 2013; Schnieders 2006). Recently research has intensified as it is being considered as a configuration approach for SaaS solutions: software programs that are shared across using organization and are made specific through configuration (Romeroa and Vernadat 2016). We extend the scope in our approach by applying this technique across the full range of business design entities (not only products but customers, partners, channels, etc. as well). In addition, we apply feature modelling to the full scope of operating model components that populate the operating model.

Achieving adaptability

Using the two basic techniques of partitioning/-componentizing and feature modelling, adaptability is achieved as follows for the three types of changes at the three layers of the solution model:

Business model: Foundational changes create new instantiations of business design entities. They can identify new value propositions, new markets, new customer audiences, new products, new production methods, etc. *Routine* changes create new feature configurations of existing business design entities. For example: 1) new and/or extended versions of the eligibility policy of an insurance product that assesses if a customer can apply for it, 2) new and extended versions of lifestyle profiles of customer that are used to target sales and marketing efforts, 3) an updated set of attributes that define the usage of an interaction channel by a supply chain partner, et cetera. *Transitional* changes are a mix of new instantiations of business design elements and extending feature configurations of existing ones. For example, addressing a new customer segment with a product with extended features.

Operating model: Foundational changes create or extend partitions and add or extend operating model components in these partitions. *Routine* changes create new feature configurations of operating model components. Where feature modelling of the business model focuses on the business design, that of the operating model focuses on the operating environment that supports the business. In terms of an example from physical manufacturing: where the color and engine type of a car are business design features of the product, the script that an assembly robot requires to mount a motor in a car on an assembly line is a feature of a function of a component in the operating model. *Transitional* changes are a mix of new operating model components and extensions of existing components.

Technology model: The modelling at this layer is comparable and follows to a large extent the modelling of the operating model. This includes

Vol. 16, No. 5 (2021). DOI:10.18417/emisa.16.5

partitioning and componentization. The adaptability modelling needs to take the metadata provided by the technology into account. Feature modelling provides an abstraction across the metadata of the specific technology as provided by the Business Configuration Center.

Business Configuration Center

To support feature modelling across different implementation technologies with their own metadata, we propose a separate software component, the Business Configuration Center (BCC) (at the far left in Fig. 3). Feature modelling through the BCC envelopes the large variety in metadata in implementation technologies with a standard approach. At the time in our research when we developed this concept it had not been described in this full form (Bruls et al. 2016). Since then similar approaches have been claimed as patent and described in the literature as well: for example configuring of solutions with different sets of metadata (Bryan and Kavantzas 2017) and academic research to configure PLE features (Horcas et al. 2017). In our proposal, the Business Configuration Center (BCC) acts as the central console that performs integrated configuration of features of business design entities in the business model and of components in the operating model (see Fig. 4). After administration of adaptability at

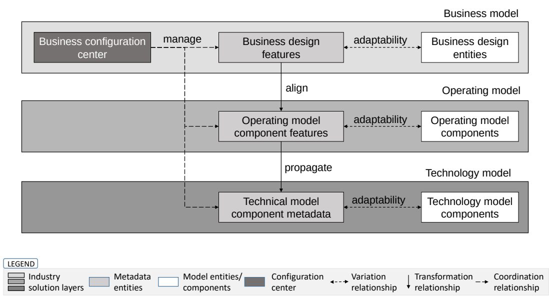

Figure 4: Integrated Feature Configuration

the layer of the business model, the BCC can be used to administrate the features that configure the production infrastructure of operating model components and the metadata that configure the technology model components (middle and bottom of Fig. 4. See for our proposed method to perform alignment Sect. 6.3.2. The propagation to perform the changes to the production infrastructure that match the newly selected features from the business design can be automated in the BCC. If this automation is too complex (particularly if alignment of production infrastructure and business design entities is not perfect), then separate configuration facilities can be established for IT personnel to configure technology model components (bottom layer) directly.

Change paths

	Routine adaptability	*Transitional adaptability*	*Foundational adaptability*
Business model	(Re)configuring existing features of business design elements	New/extended business design elements with configurable features	New value propositions with new business design elements
Operating model	(Re)configuring existing features of operating model components	Extended partitions with new/extended operating model components	New partitions with new operating model components
Technology model	(Re)configuring existing technology model components	Extended partitions with new/extended technology model components	New partitions with new technology model components

Table 2: Type of change paths

Tab. 2 summarizes how for the different levels of the solution model the combination of the adaptability entities we propose (features, components, partitions) can support the three types of adaptability (foundational, transitional and routine).

6 Analysis and Engineering Method

This section details the Analysis and Engineering method introduced in Sect. 4. Fig. 5 depicts this method as two separate semi-circles: one focused on analysis (outer semi-circle in Fig. 5) and one on engineering (inner semi-circle in Fig. 5). They center on the solution model that is produced (bottom center part in Fig. 5). The three phases through which the analysis and engineering proceeds are depicted in Fig. 5 across the semi-circles. The activities that investigate adaptability are specific for each phase. They assess the impact on business, architecture and design structure using three maps:

Vol. 16, No. 5 (2021). DOI:10.18417/emisa.16.5

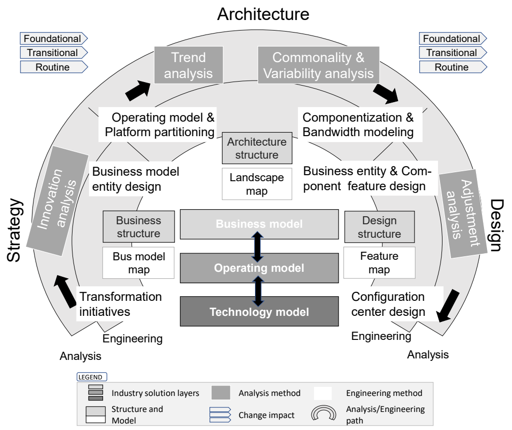

Figure 5: Analysis Method

business model map, landscape map and feature map (the inner layer of the inner semi-circle in Fig. 5). We introduce the analysis and design techniques that are detailed in follow-on subsections. (For an overview of the extant literature, and how we build on that see Appendix A.2).

In the strategy phase, the analysis starts with considering the transformation initiatives that the enterprise has defined, and assesses their impact on the business model map. New value propositions, changes to the product portfolio, customer segments, etc. are designed. And new and changed sets of processes, activities and resources required to support these. In the architecture phase, the analysis investigates the transformation impact on the landscape maps of the operating model, adding a broader view that includes all relevant business and IT trends. Partitioning and componentization modelling is performed to arrive at the best landscape layout, and commonality and variability modelling to establish optimum bandwidths of components. In the design phase, the adjustment analysis investigates the impact of changes on the feature maps of the business and operating model. Feature design is performed to establish new or updated features, including integration of features in the business configuration center and the translation into technology specific metadata.

As depicted in Fig. 5, in each phase the full set of expected changes from routine changes through transitional to foundational changes is elaborated. Analysis may start in every phase at every layer, it may be performed top down or bottom up. For example, assessing an innovation of the business model and following the impact down through to the operating model and technology model, or vice versa assessing the potential of a technology at the technology model first and from there to operating model and business model. By applying the identified impact assessment methods, top-down and bottom-up alignment of adaptability from the business model to the technology model is achieved.

As Tab. 3 illustrates, not all models are used equally in each phase.

	Strategy	*Architecture*	*Design*
Business model	X		X
Operating model		X	X
Technology model		X	X

Table 3: Use of models across phases

In the Strategy phase the dominant model is the business model, as it allows to assess in which part of the business the impact is largest. As the Architecture phase is about creating high level construction structure, the construction focused operating model and technology model are the dominant models. The emphasis in the Engineering phase is on organization and alignment of features across business design elements and operating model components, with the metadata of the Technology model providing enabling options.

Appendix C includes as a running example, an expository instantiation of a tax agency that is going through a series of business model innovations, some transitional, some foundational. To illustrate the analysis, the impact of these planned innovations and other prevalent trends on operating model, operating model components and configuration is 'calculated'.

Vol. 16, No. 5 (2021). DOI:10.18417/emisa.16.5

6.1 Strategy phase

The strategy phase assesses the impact of transformation initiatives on the business model. The selected model on which analysis and design activities in strategy phase center is the one that underlies the business model innovation research according to (Giesen et al. 2007). We organize Giesen's model into a map with a positioning region, a portfolio region and a construction region as depicted in Fig. 6. This organization in regions is a specific extension of our research. The first

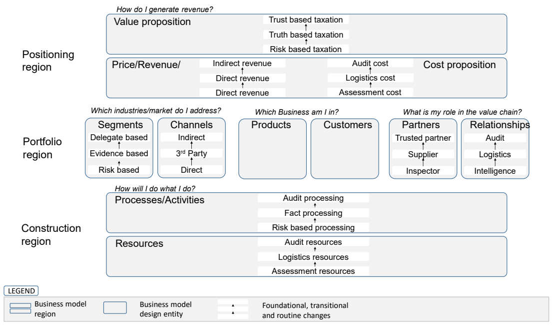

Figure 6: Assessment of impact on business model map

reflects the strategic propositioning view (why the business is attractive: the value proposition, price/revenue structure and cost structure), the second the business portfolio view (what the business offers to whom and with whom: the products, the customers – their segmentation and the channels across which they are reached, and the partners and their types of relationships), the third the construction view of the business (what runs the business: the processes and activities executed and the supporting resources).

The portfolio region is the level at which we will apply family modelling and feature modelling of the business design (see Sect. 6.3.1). The split between portfolio and construction region allows separate assessment of impact on business and operating model and alignment between these. The construction region allows us to connect with our analysis at the level of the operating model (activities and processes bundled into components) and technology model (resources). It will be detailed in the next Architecture phase into a full operating model. Innovations can result in routine, transitional and foundational changes – three examples of which are shown in Fig. 6, two transitional ones and one foundational. See for an explanation of these cases the running example in Appendix C.1, and for a detailed description of their impact on the business model Appendix C.2.

6.2 Architecture phase

The Architecture phase performs trend impact assessment to identify the impact of business and IT developments on the operating model and technology model. It designs the adaptability structure by identifying partitions and operating model components, performs commonality and variability modelling of operating model components and technology model components and establishes optimum bandwidths.

6.2.1 Trend impact assessment by partitioning and componentization

Impact of trends is assessed by projecting them onto the operating model and understanding which partitions and components need to be extended or created. This process requires a representation of the operating model that reflects the key organizing perspectives that the assessment is concerned with. Operating models are represented visually as landscape maps, with the two dimensions of the map available to represent organizing perspectives. Two dominant ways exist to view this organizing perspective. The first takes a usage perspective focusing on the functions components expose when considered from the enterprise value chains; the second takes a construction perspective focusing on how components are constructed, and boundaries are created with respect to each other. As our focus is assessing the impact of trends on the adaptability of the solution construction, the partitioning approach that we apply is a topological approach that uses construction proximity between partitions and components to organize them. This produces landscape maps as depicted in Fig. 7 that compare to those produced by Torre et al. (2013). The dimensions of the landscape views that we introduce are specific for our method; they have been selected to allow optimum isolation of trend impact. The horizontal dimension plots partitions and their operating model components according to an outside/inside gradient, that reflects the structure in enterprises that has emerged over the years while enterprises opened their production-oriented IT environments to integrate external processes and partners. Our model includes that part of the external ecosystem (the leftmost pillar) that is directly relevant for the enterprise operations. The vertical dimension plots partitions and their operating model components according to their role (interactive work, production flows & functions, production assets) in the enterprise construction paradigms that have surfaced over the years: 1) the introduction of interactive graphical user interfaces, followed by task-based workflow, workspaces, etc.; 2) the introduction of batch production systems, followed by transactional systems, process control and case processing; and 3) the introduction of assets in various forms (structured databases, document repositories, intelligence information) and physical production control.

The extent to which partitions are detailed into business functions depends upon the specific nature of the industry. Physical goods manufacturing industries will detail the inward oriented functions and assets that support physical production to a large extent, whereas relationship focused industries will create more detail in the outward looking relationship management tasks. For an administrative authority (like the tax agency from our running example) the detailing focus will be on the information and intelligence aspects of tasks, functions and assets, with equal emphasis on both the inward and outward oriented aspects.

For partitioning and componentization techniques, we leverage standard enterprise architectural methods that create coherent regions of autonomous functions isolated with respect to each other (Lankhorst 2017; Sessions 2008). As functional cohesiveness is an important driver, the partitioning structure of the technology model follows that of the operating model to a large extent (see the example in Fig. 10). Differences may occur, for example, where implementation choices collapse or split technology model components. For example, a package solution supporting multiple business functions from the same instance, or a single component split into multiple micro service partitions.

Where the innovation use cases in business model innovation assessment focused on formulated strategies, the analysis at the level of the operating model now zooms out to assess the impact of the full set of business and IT developments that are considered relevant. For every trend that is expected to occur and that may become an important shaping driver of the operating model, the impact is assessed: it may produce new, growing and shrinking partitions and new, extended or decommissioned components. The example in Appendix C.3 illustrates the analysis for the running example of the tax agency.

6.2.2 Component commonality and variability modelling

Operating model components are coarse grained constructs that envelope both the processes and activities that are part of the Construction layer of the Business model (see Sect. 6.1), as well as more detailed entities such as processes, services, functions that are used in design and engineering methods. Although differences exist in the exact definition, they compare to and are defined with the same intent as similar coarse grained constructs identified with different names such as for example architecture building blocks in enterprise architecture methods (TOGAF 2021), business components in business strategy methods (Flaxer et al. 2005; Harishankar and Daley 2011; Pohle et al. 2005) or capabilities in strategy research (Caas 2021). These are all used to translate strategy into construction structure without drowning in detail.

To model the *common* contents of operating model components we build on componentized

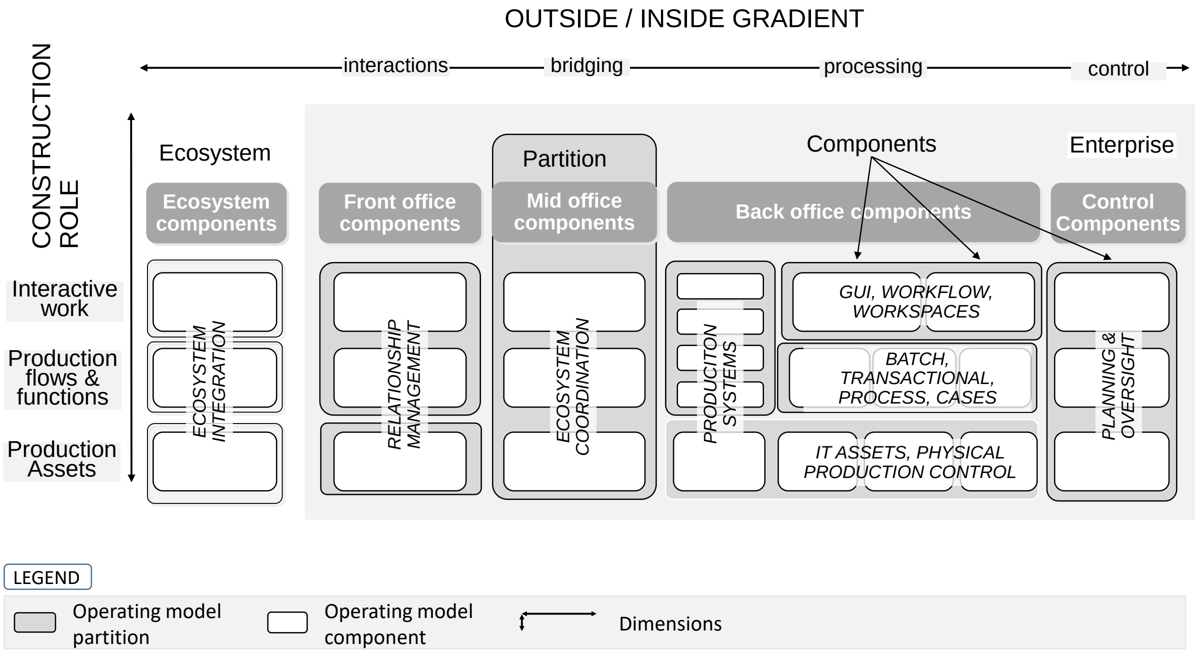

Figure 7: Landscape map of the operating model: partitions are represented by darker horizontal or vertical rectangles, components by white boxes

frameworks that have adopted a Service Oriented Architecture (SOA) construction view. These frameworks have developed standardized components for specific industry contexts (Bruls et al. 2010; Evernden 1996). They use model driven methods (Arsanjani et al. 2008; Levi and Arsanjani 2002) to specify the generic reusable processes, services and information that components deliver. To model the *variability* scope of operating model components we build on the development of a Common Variability Language (Haugen 2013; Haugen et al. 2008). This is a formal language that introduces variation points in a base engineering model. It specifies variations as a set of configurable features (the variation spec), with selection governed by a constraint language that can be determined separately. See the running example in Appendix C.4 for an example.

As explained in Sect. 5, feature modelling has its roots in the field of product line engineering - in which families of products are equipped with sets of features. Carried over to IT, it has been applied to domain specific languages to add variability to the commonality that these models represent. In this paper, the domain is an industry branch.

6.2.3 Optimum bandwidth

The configurability of operating model components is implemented by the adaptability providing technology incorporated in the underlying technology model. The selection of the technology model components and their scope of adaptability is the result of a design process that considers a combination of operating model and technology model components. Together these deliver a bandwidth of adaptability within which the system can be reconfigured. The optimum bandwidth is that point where the cost of the complexity of constructing and operating a single combination of operating model and technology model components with broad scope exceeds the costs of a combination of multiple operating model and technology model components with a narrower scope but with increased interfacing needs. Cost should be compared with respect to outcome in terms of the enterprise's business operations and strategy. Winter (2011) coins the term economical artifact engineering to reflect this process of balancing investment cost in terms of outcome. Optimum bandwidth of adaptability of components is based on these generic trade-offs between scope and construction cost but stands out in our approach with more emphasis than elsewhere (Engel et al. 2017).

The 2nd Example from practice of a judiciary agency in Appendix B already illustrated the large risks involved with inappropriate choices: an operating model component designed to support strict process driven handling of court cases failed to support the case-based handling of complex court cases. The root cause was the fact that the underlying technology model technology relied upon explicit process orchestration only. See Appendix C.5 for examples of various drivers that govern trade-offs.

6.3 Engineering phase

The Engineering phase performs the feature design of both business model entities and operating model components, organizes entity and component instantiations into families and aligns features across both layers. Also, the overall design of the feature map that is managed from the business configuration center is performed.

6.3.1 Family modelling, Feature design and alignment

As explained in Sect. 5, feature design is one of the two key adaptability providing techniques that we rely on. Product line Engineering designs a product portfolio out of families of products with selectable features that are used for configuration of adaptability. (See Appendix C.6 for an example of family modelling). In our method we apply feature modelling across the full range of business design entities and the full scope of operating model components that populate the operating model. By aligning adaptability bandwidths that sets of features create both in the business design and in the operating model components, an integrated enterprise structure is designed vertically across the B2IT stack. A full design exactly matches the adaptability of the production infrastructure with that of the business design. Such an alignment considerably reinforces the effectiveness of the adaptability design, as it can propagate changes in the business quickly into changes in the operating model. Although this alignment is standard in manufacturing when applying PLE, it has to the best of our knowledge not yet been applied to the field of industry solutions. As a manufacturing example, consider the production line in an automobile factory; it contains production machinery that performs production tasks (like motor assembly, chassis construction, and painting). The scope within which the machinery can be adjusted should match the variation in features of products parts. For example, the reach and range of movements of the robot that welds car chassis parts together should match the variations in sizes of chassis parts of the family of cars that are produced on the production line. The concept of optimum bandwidth is illustrated by the fact that a certain production line will be able to build compact cars and perhaps small vans, but larger vehicles with different designs (for example SUV trucks) will exceed the adaptability of the production units.

Bandwidth alignment assumes that central control can be exercised through an organized business model. This will not hold in those cases where multiple loosely coupled partners contribute, for example across an ecosystem or in an enterprise across business units with large autonomy. In that case feature modelling of the operating model components proceeds more independently. The partners that supply operating model components tune in to trends without explicit orchestration from a business model. The partners that consume services will need to apply more integration effort to assemble a solution without the up-front control over the bandwidths. Full bandwidth alignment in those cases where it is possible can have downsides as well. It can create entrenchment in specific setups that is difficult to overcome when a change occurs that is outside the bandwidth.

Although trend impact assessment aims to anticipate the changes and prevent entrenchment, it may underestimate the contingencies and turbid developments that exist in real life environments. Loosening the bandwidth design then must ripple through to every related part of the enterprise at the same time, with potentially considerable effort and delays in time to market. This is the risk of too much alignment (Tanriverdi et al. 2010). More organic forms can be supported by less stringent alignment of the business model and more open bandwidth designs that are not too narrowly casted as a kind of insurance premium against incorrect foreseen change. These may be better equipped to support environments with a lot of unexpected change, and to exploit unexpected innovation options (Tanriverdi et al. 2010). In such a set-up operating model component variability modelling is treated as a potential without aspiring strict alignment.

6.3.2 Business Configuration Center

As explained in Sect. 5, we propose centralized business configuration as the approach to perform vertical integration of configuration across the B2IT stack. As indicated in the previous Sect. 6.3.1, feature designs at the level of business model and operating model need an approach to alignment that in this way also ripples through to the BCC configuration space. It creates a modular and well-organized set of features with limited transformation needs.

A second objective of the BCC is to put the control of enterprise change into the hands of the business users themselves. It allows business users to manage adaptability in a language familiar to them. It will be part of a broader change control organization that will handle non-IT aspects as well (for example hiring new personnel to perform a newly introduced step in the enterprise production process. Metaphors can help to create a feature language that is intuitive to persons with a business background and that they can relate to; they can be misleading as well and should be chosen with care. For example, consider an information input pipeline that feeds a repository, with data packages retrieved from there during business transaction processing. Key characteristics are the uncoupling of the information pipeline from the processing and the ability to reuse data packages across multiple business transactions. A helpful metaphor to explain this is an assembly line for product parts that stocks a warehouse, with parts later picked up to produce several types of products. See Appendix C.7 for an example of how a BCC is used in the configuration of a Single Window component.

6.3.3 Integrated design of feature map

In addition to vertical integration across the B2IT stack as explained in Sect. 6.3.1, features of individual business model entities and operating model components need horizontal integration across the business model and operating model as well. An overall feature design at this level will ensure that future routine changes can be handled coherently. Integrating features requires the introduction of a feature combination language, which has been developed in various forms – with hierarchical and orthogonal variants as most popular ones (Budiardjo and Zamzami 2014; Reinhartz-Berger and Figl 2014; Reinhartz-Berger et al. 2017). The constraints that govern feature combinations are expressed either in the hierarchical model on the junctions (for example, a car model has at least two doors and at least one color), or in the orthogonal model as separate constraints across the full feature space.

The design artifact we propose to manage this integrated feature design is a feature map as depicted in Fig. 8. We opt for a loosely organized orthogonal approach in which we define use cases on top of the features to validate both the consistency and integration of features artefacts across major flows through the operating model. For example, in case of a use case that adds a new product, actions need to be taken across many of the operating model components and associated features in the map. The base model on which the feature map is 'drawn' is the partitioned operating model, populated with operating model components. For each component we decompose into

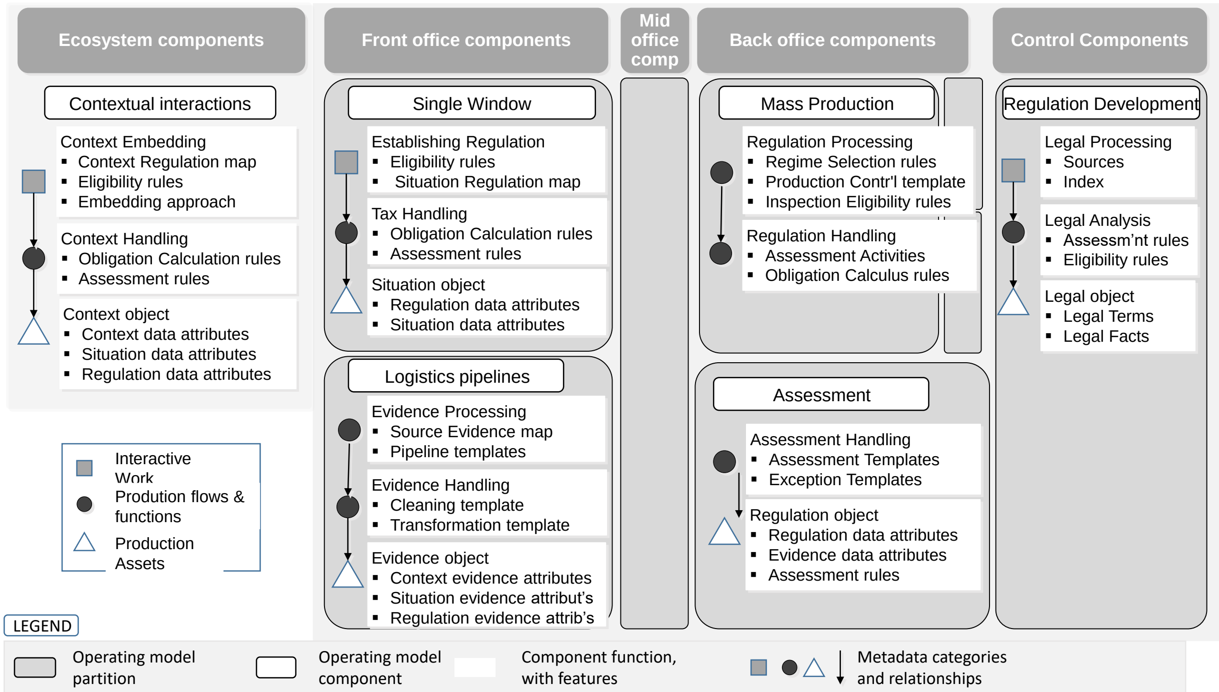

Figure 8: Feature map

Interactive Work, Production flows & functions and Production Assets (the vertical dimension of the operating model – see Fig. 7). Features are identified per function, with use cases establishing coherence across functions. Coherency of the business design can be assessed in a similar manner by using the business model to 'draw' business design feature models.

The implementation of this feature map that uses use cases on business model and on operating model to assess coherence, is specific for our method. See Appendix C.8 for an explanation of the examples in Fig. 8.

7 Evaluation and Discussion

We have evaluated and refined (see Sect. 2) the artifacts during the development cycles based upon their utility that surfaced from experiences in practice. Evidence is presented in Appendices A, B and C. Appendix A summarizes how we build on the extant literature. The examples from the practice of the authors in Appendix B illustrate the key drivers behind our approach. The expository instantiation in Appendix C illustrates its usage. How we meet requirements is illustrated in the subsection Meeting Requirements in this section. Together this provides for the grounding in the literature, the empirical grounding and the internal consistency that (Goldkuhl 2004) recommends for new artifacts.

Meeting requirements

The three layers selected for the solution model that build on the extant literature cover the design of the business model, the operating model and the technology model. Together these offer the minimum viable complexity (Req 2) that can support the adaptability needs of the large enterprise across business, operations and technology. The three phases selected for the analysis and engineering method cover the strategy, architecture and design perspective. Together these offer the minimum viable complexity (Req 3) that can support the adaptability needs of the large enterprise across three levels of change: foundational, transitional and routine (Req 1). Key features of our approach match requirements as follows:

- The central adaptability enabler is the architecture structure of the operating model that consists of partitions and components. This limits side effects when absorbing the changes that disruptive trends bring (Req 5)
- Component configurability and variability design supports optimum bandwidths of change (Req 6) for routine changes
- The integrated modelling across layers that uses dependency relationships across a limited number of modelling elements (Req 4), ensures that the changes can be aligned with those in the business model design and the technology model
- Through technology agnostic feature modelling, simple enough configurability is achieved (Reqs 7, 8). This isolates the adaptability design from the specifics of the metadata technologies in the technology model.
- With the help of metaphors, the feature modelling can be further simplified to allow control to rest with the business owning organization.

Key assumptions

Two key assumptions underlie our approach of 'designed adaptability': that change can be foreseen and that it can be prepared for by pre-'wired' operational structures. There is certainly a limit to the extent that an enterprise can foresee and prepare for changes using the adaptability approach we propose. Also, our approach is no guarantee that every change within that horizon will have been accounted for. However, we do not see alternatives that are available other than foresight. Certainly, adopting adaptability techniques such as the two selected in this paper do provide by themselves a level of flexibility that will be able to cope to some extent with unforeseen changes. However, the spectrum of changes that we discussed includes those that are of a magnitude that cannot be left to merely generic adaptability approaches without the attempted foresight and assessing the impact on the operating model design.

The second assumption that change can be prepared for by 'pre-wired' operational structures rests on the two adaptability techniques identified in Sect. 6.2.1: architecture partitioning and componentization, and feature and metadata modelling. The notion that an architecture adaptability technique such as partitioning and componentization can contribute to a future proof solution is common knowledge in architecture practice. And the notion that an engineering adaptability technique such as feature and metadata modelling can support a large adaptability range in an industry solution is becoming a touchstone for SaaS paradigms. As enumerated in Appendix A there are certainly other techniques then the ones that we propose. For example the usage of mediating layers that bridge between old and new (to allow legacy environments to quickly offer new services to new environments), model driven development of systems that focuses on the model for extensions (Blair et al. 2004; Hinkelmann et al. 2016; Szvetits and Zdun 2016), or self-adaptive mechanisms such as approaches that range from genetic to artificial intelligence-based algorithms that can react to specific new patterns (Rodrìguez et al. 2016). These techniques can be applied in addition to what we proposed in this paper to gain additional degrees of flexibility and extensibility. However, to achieve a 'simple enough' approach we have limited ourselves to the two that can support the mix of changes: partitions to hold full new extensions for foundational changes, components to model new and changed building blocks for transitional changes, and bandwidth-based configurability to support routine changes.

Applicability of our approach

The impact of foundational changes is assessed as extension of an existing operating model. Many of today's trends as identified in Sect. 6.2.1 fall into this category – and can be analyzed using the method we introduced. There are, however, changes that are so disruptive that an operating model extension will not be enough. These changes may require a full new business model and a corresponding full new operating model. Consider for example the impact that digital photography had on KODAK company (Haftor 2015), with a business model around physical film, to the radically different business model that Instagram developed with revenue based on advertisements around published digital images. Examples of similar challenges to existing enterprises are those brought about by other digital companies such as GOOGLE, SPOTIFY, NETFLIX, UBER, and AIRBNB (Haftor 2015). Although our analysis method does allow us to assess the construction of a new operating model under these trends, the analysis will need to proceed much more at the level of the business model first. This implies developing multiple scenarios with views of industry wide competitive positioning and value propositions and assessing at a high level the implications for processes and resources needed. Once an attractive variant is established, the operating model design can proceed again with a largely blank canvas, reusing and integrating existing parts of the enterprise's operating model where possible.

Creating the integrated structure of the overall architectural models, the integrated feature design as included in the BCC, and performing the integrated analysis and engineering activities may appear at crossroads with agile and distributed ways of working (Lankhorst 2012). Distribution, however, is certainly possible. For example, by introducing domain concepts in the BCC, allocating responsibility for partitions, operating model components and parts of the technology model to different architecture, design and development groups. However, for designed adaptability to be effective at the enterprise level, the integration issue will require resolution through a governance approach that imposes some central coordination. This may be provided by, for example, enterprise architecture management (Lankhorst 2017).

Although we presented the analysis method in a sequential top down narrative, there is no reason for executing it like this. Entry points are available at every layer and impact assessment can be performed bottom up as well. Particularly if the impact is large, then feedback loops are required that revisit the impact at the various layers.

Emergent versus designed

Several trends are appearing that are expected to bring a (r)evolution to industry solutions that we summarize as 'emergent' adaptability. One is the increased importance of ecosystems and 3rd party platforms as a foundation for business (Basolea et al. 2015). The control over the business model is much less strict in these setups, and service collaborations across partners are expected to emerge rather than to be designed (Barros and Dumas 2006; Zimmermann et al. 2014, 2015). The second are the analytical and cognitive technology trends that allow new behavior and insights to be 'self–learned' or 'trained' rather than designed. This allows change to become emergent – to develop without up-front design. The developments in the field of analytical and cognitive technology are producing intelligence components that are increasingly integrated real time into business processes and functions, including those in the front office. This diffuses sensing, decision making and responding from a back-office product design focus towards other areas with increasing local autonomy. For example, a customer interaction in which analytical technology real time identifies the previous interactions of (similar) customers and in which cognitive technology accesses a knowledge base with learnt schemes extracted from that to decide on the best propositions. A development that is a considerable step beyond mass customization and has been coined the Individual Enterprise (Marshall et al. 2014) to reflect the highly personalized experience.

The BCC is a key component in the designed approach to adaptability. At the same time with respect to emergent trends, it is the hybrid step towards that. It bridges between a world in which required adaptability can be analysed and can be frozen in variability schemes that can be programmed and a world where intelligence is fluid and very contextual and the result is different every time it is applied. The BCC hub can be considered the command center in the 'enterprise mind' that controls the procedural part of the enterprise. In a parallel to human beings: consider it that part of the brain that controls – outside of the consciousness focus (where emergence resides) – the standard responses to the environment (drinking coffee, walking a specific route towards office, recognizing friends and adopting contextual behavior). These routine procedures are highly efficient and need limited resources. New routine patterns can be 'programmed' in the BCC as they emerge repeatedly from the intelligence parts.

Future work

Extension and deepening of our research is required in various areas. Firstly, gathering additional evidence that the approach will hold its ground in practice. Secondly detailing insights in the design of the artifacts: solution model, adaptability techniques and analysis and engineering method. For example, detailing the dimensions of landscape models, formulation of feature models and constraints, and translation techniques required in a BCC to map features to technology, etc. Thirdly assessing industry specific variability across artefacts, for example, the extent to which operating models and bandwidths of adaptability differ. And finally, the integration with other adaptability paradigms, such as agile development approaches, self-adaptive and emergent approaches, etc.

8 Conclusion

Adaptability is at the center of what allows enterprises to adjust their business and operating models in highly turbulent environments. A rationale base to entrepreneurial activities requires anticipation of change and coping with it. The solution model and analysis and engineering methods that we have provided are shaped at its core by the idea of mastering change with a designed approach of limited complexity. It allows to address today's wicked mix of changes and yet produce results that are understandable for and explainable to enterprise decision makers. We have inherited from a broad set of existing methods to design the adaptability. The approach that we propose constitutes a layered set of models that can be used to assess the impact of variations at multiple layers. Multiple alignment mechanisms extend and integrate across these layers.

The metaphor that matches designed adaptability is that of mass customization – it structures configuration artifacts in the business configuration center and provides business planners with the frame of reference from which they can reason about adaptations themselves. We have previewed emergent adaptability as a new approach that will create new parts in the operating model and technology model. Trends such as cognitive and analytical computing and social collaboration result in much more individual and group experiences than the mass customization metaphor.

The multitude of choices for operating environments and the inherent complexity that is introduced by the impact of various types of change illustrate that large enterprises are amongst the most fascinating artificial phenomena. Adaptability is a central theme that is a foundation under the vision of the enterprise as a complex adaptive system (Onix et al. 2017; Schilling et al. 2017). It has not received a great deal of attention as a separate topic in research. We encourage research endeavours that further the science of adaptability.

A Appendix - Building on the extant literature

We summarize in this section the research fields that have inspired the assembly of our solution model and the analysis and engineering method and how we build on them. First, we review related work in adaptability providing techniques in the solution model, then related work in adaptability focused analysis and engineering techniques. Relevant literature has been surfaced using the term 'adaptability' and related terms ('modifiability', 'variability', 'composability') as broad search themes with deepening in the various layers of solution model ('componentization', 'configuration', 'model based', 'wizards', 'variability language', 'product lines') and method ('business model', 'product architecture', 'capability architecture', 'enterprise architecture', 'software architecture'). Given the broad scope of the paper, our survey is representative rather than exhaustive. Also, given our focus on designed adaptability, we have not covered development practices (such as agile development practices and model driven development) in large detail (Lankhorst 2012). (See Sect. 7 for positioning these).

A.1 Related work on Solution Model

Adaptability providing techniques in business and IT solutions have been an important topic since the resistance to change of the first monolithic production systems became clear. We review the extant literature, and summarize at the end how our integrated solution model builds upon it.

Operating model

Strategies to address the resistance to change at the level of the operating model have used two major approaches. One is the partitioning and modularization of business functions that allows separation of concerns and limits side impact of changes. It is established best practice in enterprise architecture (Lankhorst 2017) and software engineering (Mannaert et al. 2012), as well as in physical building construction and design of production machinery (Engel and Reich 2015). Second is the modelling of the generic and variable aspects of business operations (Harishankar and Daley 2011). This type of research has been triggered by global organizations with local differences in business processes and business services (Ayora et al. 2015; La Rosa et al. 2013) and other architectural constituents (Rurua et al. 2017). Examples include reference modelling that derives situated models from generic models (Brocke 2007), service-oriented modelling methods that identify generic functions and their variability (Arsanjani et al. 2008; Levi and Arsanjani 2002) and variability modelling that creates generic models to represent variability across arbitrary entities. The latter has resulted in the development of a Common Variability Language (Haugen 2013; Haugen et al. 2008) that was then considered by OMG as a standard, and now with increased interest from the SaaS industry (Romeroa and Vernadat 2016).

A more recent theme is research into self- adaptive systems. This started with the introduction of the concept of autonomous computing (Kephart and Chess 2003) defined as computing systems that can manage themselves given high-level objectives from administrators. Componentization has been studied as a foundation for this behavior (Stoicescu et al. 2017). This research wave initially mostly addressed simplifying the management of a system (installing, configuring, operating, optimising, etc). Although it branched out into other areas (like robotics, or device control) it mostly stayed at the level of the infrastructure and resource base – however with limited practical applications yet (Weyns 2019). Interpreting runtime models and changing them to create self-adaptive systems has been a line of research that has built on model-based development of systems (Blair et al. 2004; Hinkelmann et al. 2016; Szvetits and Zdun 2016). The origination of ecosystems and industry platforms with solutions that need to be assembled using services made available by different participants, has seen a research field develop that investigates the adaptability of these services and the automated assembly of solutions (Zimmermann et al. 2014, 2015). This research field has focused on approaches that range from genetic to artificial intelligence-based algorithms (Rodrìguez et al. 2016).

Technology model

The IT provider industry has invested heavily in the concept of generic adaptable technology solutions that can be leveraged across multiple enterprises (Romeroa and Vernadat 2016). This includes packaged applications for specific industry segments with built in configurability and customization options, and middleware stacks with modelling environments that allow to support different types of models such as business processes and information management processes. Academic research has focused along similar lines on domain and context sensitive models with middleware designed to interpret variations (Li et al. 2015; Morrisa et al. 2015). Research into how to build configurability into these generic solutions has recently intensified as software is increasingly delivered as cloud-based Software as a Service (SaaS). In this delivery model a single instance of the software is shared across enterprises (Romeroa and Vernadat 2016) and configurable middleware is used to support the required variability (Van Landuyt et al. 2015).

Metadata

To integrate diverse sets of metadata (Bryan and Kavantzas 2017) across technology solutions of different origin, various formats have been considered such as business rules (Keddis et al. 2015), knowledge graphs (Amoui et al. 2012; Mongiello et al. 2016), and configuration wizards (Al-Shardan and Ziani 2015). Product Line Engineering (PLE) techniques that originated in manufacturing, use selectable features to configure an enterprise's product lines (Horcas et al. 2017). This approach has been applied also to configure IT solutions that support the business operations through which the product is produced (Pohl et al. 2005). Initially focused on single artefacts such as process models and services (Cognini et al. 2015), the configuration requirements of SaaS packages has broadened this to the full set of artefacts that these solutions contain (Al.Busaidi and Kraiem 2017; Mazo et al. 2014). The scope is extensive as well, covering multiple product lines to configure the full set of production processes (Holl et al. 2012), and multiple levels of solutions such as at industry level, at enterprise level and at individual level (Rabiser et al. 2009).

Summary and usage of existing approaches

In summary partitioning and componentization and variability modelling are major approaches to adaptability at the level of the operating model, with at technology level various technologies that support configurability such as application packages and middleware that all have their own approach to configurability. Research into integration of metadata has researched various formats with PLE feature modelling broadest in scope. We build on this literature, by extracting two simple adaptability techniques that we will apply across the B2IT stack. We use partitioning and componentization techniques to structure the solution at both the level of the operating model and the technology model, and PLE feature modelling to configure solution elements at both the level of the business model design entities and operating model components. The proposed implementation of the feature modelling in what we designate as "Business Configuration Center" is a specific proposal from our research. It propagates PLE based metadata across the technologies such as packages and middleware driven solutions at the technology level Our assumption of designed adaptability reserves self-adaptiveness on top of these techniques for follow on work.

A.2 Related work on analysis and engineering method

To assess and design impact across the B2IT stack, integrated analysis and engineering across it is required. We review the extant literature in this area, and summarize at the end how our analysis and engineering method builds upon it.

Business models

Research into analysis of business models has focused on understanding alignment between competitive position, industry value chain position, core business entities such as products, customers, partners and channels, and core production entities such as processes, activities and supporting resources. Business model innovation research (Giesen et al. 2007) is covering the spectrum between gradual and disruptive innovations (Girod and Whittington 2019). The latter reflects the simultaneous impact of many trends that result in complex reconfigurations with equally drastic impact on the operating model design (Alvertis et al. 2015; Evans 2017; Haftor 2015). On a more evolutionary scale, analysis methods are researched that translate variability in the product portfolio into required flexibility of the production environment (Bonini et al. 2018; Keddis et al. 2015).

Operating model and technology models
Analysis at the level of the operating model stresses the development of independent pieces of functionality that persist over time, in multiple contexts at both the operating model and technology model layers. This limits side effects of changes. Wehling et al. (2016) perform architectural variability modelling at these layers to reduce complexity. The key trade off in modularization is the size of a component that balances interface cost and the options for change (Engel et al. 2017). This trade off that leads to an optimum bandwidth for a component has also been identified in DSR-research (Winter 2011).

Modeling relationships between components requires architectural views of the operating model that plot the functions in relation to each other (Torre et al. 2013). These models are represented visually as landscape maps, with the two dimensions of the map available to represent organizing perspectives. Two dominant ways exist to view this organizing perspective. The first takes a usage perspective focusing on the functions components expose when considered from the enterprise value chains; the second takes a construction perspective focusing on how components are constructed, and boundaries are created with respect to each other. Examples of the first include Component Business modelling, a technique developed in strategy consulting to assess strategic priorities; it uses processes from the value chain and phases in the strategy lifecycle as dimensions (Flaxer et al. 2005; Harishankar and Daley 2011; Pohle et al. 2005). Campbell et al. (2017) use as dimensions usage of the resources in the value chain and organizational ownership. They do include a topological construction split into a front office (populated with external access resources) and back office (populated with internal production resources). Examples of the second include the design patterns proposed by Ross et al. (2006) that underlie the construction of an operating model. Their focus is less on partitioning details and individual components. Sessions (2008) attention is on the individual building blocks. He identifies partitioning as a fundamental technique to reduce complexity and creating so-called autonomous business capabilities, but spends little time on the overall landscape.

Modelling impact across adjacent layers in the stack requires integration of models. Enterprise architecture models (Ganczarski and Winter 2015; Harishankar and Daley 2011; Lankhorst 2017; TOGAF 2021) allow to track impact through detailed dependencies across layers (Boer et al. 2005). Coarse grained methods model business components (Flaxer et al. 2005; Harishankar and Daley 2011; Pohle et al. 2005), and high-level capabilities (Caas 2021). They are used to assess the impact of an evolving business context and the development of new business offerings (España et al. 2015), and provide better alignment then the fine grained EA models (Bērziša et al. 2015). Pastor et al. (2018) introduce for example capability-based modelling as a method to face changing business contexts that emphasizes the reuse and variability that is the topic of this paper as well. Clark (2018) introduces a strategic, tactical and operational horizon of change that corresponds to the foundational, transitional and routine adaptability perspective proposed in this paper.

Component design and development
The design of individual components is the field of software architecture. Modifiability of these architectures has grown into a rich research field that has produced methods such as scenario-based analysis techniques for change elicitation, evaluation and interpretation. These techniques are used to develop a view of the variability (Bengtsson et al. 2004; Mistrík et al. 2017)and the design guidance required to build normalized components that can be assembled to absorb the impact of changes with limited side effects (Mannaert et al. 2012). Other examples include reference modelling that derives situated models from generic models (Brocke 2007), and Service oriented modelling methods that identify generic functions and their variability (Arsanjani et al. 2008; Levi and Arsanjani 2002). Haugen (2013) and Haugen et al.

(2008) proposed development of a Common Variability Language (CVL) that was then considered by OMG as a standard (Romeroa and Vernadat 2016).

Summary and usage of existing approaches
In summary, business model methods, enterprise architecture methods and software component engineering methods are available with different types of models, and different paths to integration and different views of modifiability.

We build on this literature as follows. For strategic analysis we rely on business modelling methods that identify the essential elements of a business (Giesen et al. 2007; Osterwalder et al. 2005) but improve their connection to the level of the operating model. For architecture analysis of the operating model and technology model, we rely on enterprise architectural models in landscape mode (Torre et al. 2013).

The dimensions of the landscape views that we introduce are specific for our method; they emphasize construction aspects and have been selected to allow optimum isolation of trend impact. We use coarse grained components as the building blocks of the enterprise landscape views. For modelling them, we rely on methods from business architecture (Flaxer et al. 2005; Pohle et al. 2005) and capability related research (Caas 2021). For commonality modelling of component contents, we rely on model driven methods (Arsanjani et al. 2008; Levi and Arsanjani 2002) to specify the generic reusable processes, services and information that components deliver. For variability modelling we lean on the development of a Common Variability Language (Haugen 2013; Haugen et al. 2008) considered by OMG as a standard, and now with increased interest from the Software as a Service industry (Romeroa and Vernadat 2016). Optimum bandwidth of adaptability of components is based on generic trade-offs between scope and construction cost but stands out in our approach with more emphasis than elsewhere (Engel et al. 2017; Winter 2011).

For engineering analysis of configurability, we build on feature models from product line engineering (PLE) that have carried over from mass manufacturing (Hallsteinsen et al. 2006; Pohl et al. 2005). We extend the original concept by applying them broadly across both the entities from the business model layer and the components of the operating model. The implementation across this full space is specific for our research.

B Examples from practice

In this Appendix, we provide three examples from the authors' practice that surfaced key problems and approaches for addressing them that have inspired much of this research and that have contributed key requirements.

B.1 Example 1: prepare for impact of strategic trends by partitioning

This example demonstrates that the impact of disruptive trends can be prepared for by partitioning of the operating model (see Sect. 3.2 Requirement 5: Trend impact containment). An industry solution for European customs agencies needs to support the inspection and taxation of goods that enter a country (eCustoms 2021). The solution of the customs agency in the country of goods destination (see 9 upper right part) receives declarations in the front-office directly from the trader from its country of residency, checks them for accuracy and then passes them to a back-office declaration processing component applying customs functions such as risk assessment, goods inspections, and obligation management (such as tax collection).

The future trend is to delegate these customs functions towards partner agencies (at the point of entry in the EU, or at the country of residence of the trader) and logistics hub (harbours and airports where goods change transport mode). It reflects a business model innovation in which individual agencies and partners that play in a bigger ecosystem start to behave as a single entity, delegating the execution of the customs function to the place where it makes most sense. A similar

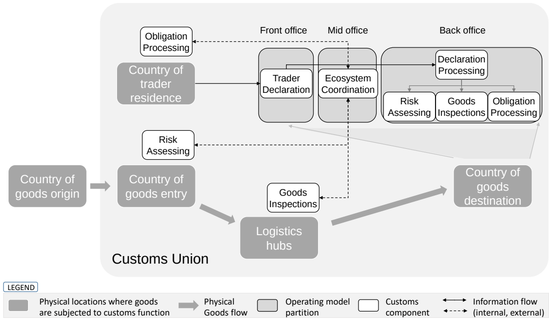

Figure 9: Example 1

trend is present in many industries: Enterprise focus on core activities and delegate non-core activities.

The trend impact analysis that investigates the impact on the operating model shows that the delegated customs functions that are executed by others still need to be coordinated by the agency that is responsible for them. To manage this coordination a new (mid-office) collaboration partition is identified in the industry solution from which this coordination is performed (already shown in the example of Fig. 9). To prepare for this change path, each individual customs agency will need to prepare its operating environment for this new partition – thus enabling implementation of this change with limited side effects.

The effectiveness of introducing a mid-office partition is evaluated by performing a root cause analysis. This reveals that where in the e-business era new partitions were appearing in the front office to interact across new channels and with new audiences, the trend towards distributed processing across the ecosystem requires a deeper integration and coordination that is run from the mid-office. This analysis implies that the proposed mid-office partition is a foundational development in the operating model that holds across industries, and that the proposed partitioning approach can absorb foundational changes.

B.2 Example 2: select the right adaptability bandwidth

This example demonstrates that the correct design of the adaptability of operating model and technology model components plays a key role in supporting the variety in business operations of an enterprise, and that incorrect choices can have serious consequences (see Sect. 3.2 Requirement 6: Optimum bandwidth).

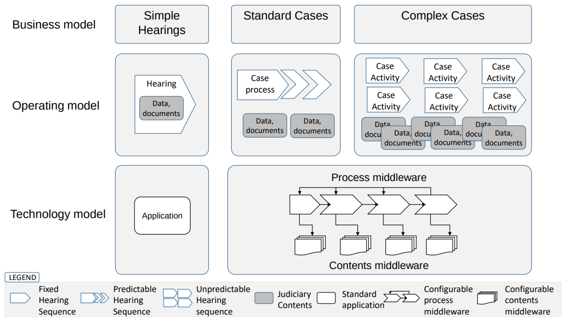

Figure 10: Example 2

As depicted in the top and middle of Fig. 10, a judiciary agency supports a spectrum of court cases (Schmidt 2007) that vary from 1) simple hearings (appeals against traffic fines – such as speeding) consisting of a single fixed procedure and standard contents, to 2) standard cases with defined procedural steps and a document store that may contain structured and unstructured contents (burglary, tax evasions, etc.), to 3) highly complex cases with unpredictable procedural steps and a large range of unpredictable contents (criminal organizations, antitrust cases, etc.). The solution for simple hearings has been developed as a dedicated fixed flow and information processing application and works well (bottom left in Fig. 10). For the standard and complex cases, a technology model has been selected with configurable process and contents middleware (bottom middle and right in Fig. 10). The configurable process middleware produces work streams that are quite rigid; they have trouble supporting the open work assignments and flexible routing that the complex court cases require. The introduction of this solution for standard cases has already met with quite some resistance (Velicogna 2007). The extension to complex court cases with much more unpredictable flows enforces a rigid regime onto users that were used to and require considerable freedom in performing work. After considerable efforts and attempts to adjust within the bandwidths of adaptability imposed by the selected middleware technology, the solution is finally abandoned.

B.3 Example 3: create understandable metadata

This example demonstrates the need for understandable and maintainable metadata to achieve the promise of an adaptable solution (see Sect. 3.2 Requirement 7: 'Simple enough' configurability).

A government agency has acquired a solution that allows citizens and enterprises to submit permit requests for activities that are regulated by law. Eg to request permission to perform commercial activities that are close to a protected state park, to extend a residential area into an area that includes infrastructure for regulating water supplies, etc. The solution has been developed using a middleware infrastructure that models all adaptable functions such legislative policies, process flows, data structures, user interactions, etc., in a single integrated semantic model. The complexity of understanding this model is such that it needs highly skilled developers to perform it, requiring separate development cycles now at the level of this metadata. This puts the business users at a distance, and results in prolonged time to market for even simple legislative changes that require complex interrelated updates across the full model. The solution once delivered does not provide the expected flexibility, and a 'no regrets' project is started to acquire a new solution.

C Expository instantiation

To illustrate our approach with examples that can prove the concept, we introduce the case of a tax agency that is tasked with collecting a variety of taxes and disbursing supplements for citizens and enterprises. The agency is performing an analysis of the impact of several planned innovations in the business model and an analysis of the potential impact of prevalent business and IT trends.

C.1 Innovation cases

The agency is going through a planned innovation in the business model, with two transitional steps followed by a foundational step. Until now taxpayers provided evidence data on their situation per tax regulation. In the innovations we consider, this changes as follows:

- *Integrated data entry across regulations, Risk based processing:* in this transitional innovation taxpayers enter data about their personal situation only once (for example information on properties owned). The data received is validated once and used for multiple tax regulations (for example both determining income tax as well as housing tax). Tax processing remains based upon assessment of potential risks of fraud by the taxpayer who submits the data.
- *Collection of evidence from 3rd parties, Truth based processing:* in this transitional innovation evidence data on the taxpayer situation is provided by trusted 3rd parties (for example providing income data by employers and financial statements by banks). The data received does not need to be screened anymore by the tax agency, and the subsequent tax processing becomes truth based.
- *Delegated compliance, Trust based processing:* in this foundational innovation taxpayers or their delegates become responsible for establishing their tax obligations themselves. Processing becomes trust based, with the help of compliance regulations that prescribe certifying procedures and audit frameworks that perform incidental controls.

C.2 Impact assessment of Innovation cases on the business model

For the three innovations that the agency is planning the impact on the business model is assessed (see Fig. 6 white boxes connected by arrows). The value proposition (top of Fig. 6) shifts from Risk-based (inspect those tax declarations with largest risk), through Truth-based (rely on independent facts submitted by 3rd parties) to Trust-based (trust selected taxpayers and intermediaries to perform a correct self-assessment). Overall the focus in the construction (bottom of Fig. 6) shifts from internal inspection of declarations with data provided by taxpayers in the back office, through receiving evidence from partners in the front office, to performing external audit processing. With each step extensions to the operating model are required that start small (allowing taxpayers to enter data across regulations and setting up new channels to receive evidence data from partners) but grow into full new sets or resources and processes/activities (allowing to set up audit frameworks, scheduling external audits, consolidating results etc). In the extreme situation where every tax collection would go through a self-assessment with audits after that, it would produce a full new operating model, and discard the current back office declaration processing systems.

From a business portfolio perspective (middle of Fig. 6), the agency remains responsible for the same tax regulations (product) that apply to the same taxpayer (customer) – no change in these entities therefore (middle center of Fig. 6). Other portfolio entities (middle left and right in Fig. 6) show impact that considerably increases with each change step. Segmentation of customers for example remains initially risk based with limited change. Next the evidence that can be acquired is an important determinant that refines the risk-based segmentation. Segmentation in the third step is completely based upon a new determinant, the trust relationship that can be established. Similar differences occur with the other portfolio entities. Partner and Relationship keep the current focus on Inspection partners (inspecting cases of fraud) and Intelligence providing relationships (identifying cases of fraud). Next, Supplier partners (supplying evidence) and Logistics handling relationships (intermediating evidence) that today already provide evidence gain additional emphasis. Then, Trusted taxpayers (performing self-assessments) and Audit relationships (auditing compliance) are completely new entities.

C.3 Impact assessment of trends on the operating model

The impact of prevalent trends on the operating model is depicted in Fig. 11. The new ecosystem components pillar absorbs major trends towards increased ecosystem integration:

- The development of online consumer platforms (1) into which tax services can be integrated and interactions become contextual (A) - with property transfer tax levied on a commercial transaction on a real estate website as an example,
- The development of new digital products (2) integrated across partners - with delegated compliance (B) as an example,
- The formation of digital ecosystems (3) that deliver 3rd ecosystem services (C) - with 3rd party evidence provisioning, tax collections, and onsite inspections as examples.

The front office components pillar includes two existing partitions, one that supports interactions with clients and intermediaries (the main entry) and one with supply chain partners (the supplier's entry). These have originated during the e-business era and continue to absorb new trends towards increased customer intimacy, product innovation and partner integration, by extending and creating operating model components:

- The development of a situational view of a taxpayer (4) – integrating multiple tax specific filings into a single window (D) that integrates required information across regulations,
- The externalization of products (5) integrated across partners – requiring new Relationship handling components for example to support compliance enabling (E),
- The extension of the logistics channels (6) that now also needs to cover intelligence received from Relationship partners - requiring new logistics pipelines that support unstructured information (F).

The mid office components pillar includes an almost fully new partition that seamlessly integrates

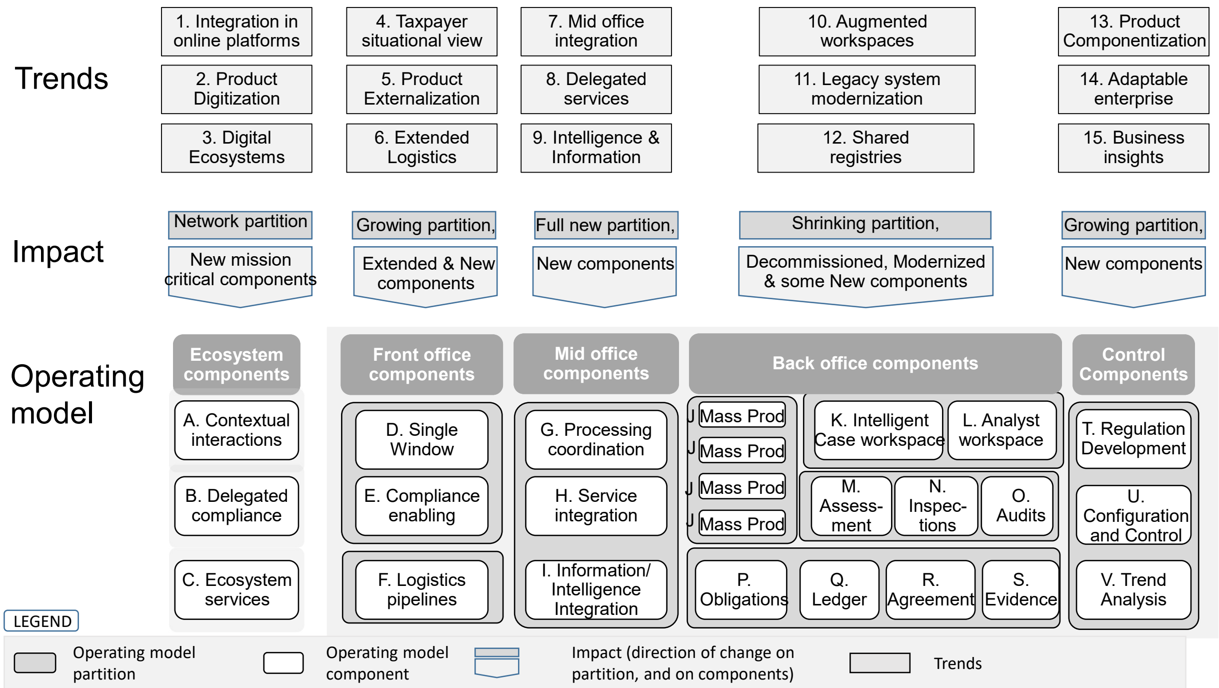

Figure 11: Impact analysis of trends on the Operating model – example

ecosystem operating model components with tax agency operating model components:

- The insertion of a new component that coordinates processes (G) that bridge between the external processing in the ecosystem network and the internal processing (7),
- The insertion of a new component that coordinates services (H) that are delegated to others due to deconstruction of the agency (8),
- The insertion of a new component that coordinates information and intelligence (I) that are the new raw material for processing (9).

The back-office components pillar includes several partitions that overall are shrinking as the agency increasingly relies on ecosystem and front office services. Task and Functional components in these areas will migrate over time to the front office and integrate with the single window:

- Current case handling (K) and inspection analyst (L) workspaces in a joint partition are augmented with intelligence and analytics technology (10),
- Assessment (M) and Inspection (N) functions are modernized (11), an Audit (O) function is added to this partition to support Delegated compliance (B),
- Existing Mass Production Legacy systems (J) in a shared partition are modernized to support operational excellence in those areas where offline mass processing remains required (11),

Vol. 16, No. 5 (2021). DOI:10.18417/emisa.16.5

- Shared Registries (12) that maintain assets like obligations (P), ledger (Q), agreements (R), and evidence (S) are included in a shared partition. They are integrated with external partners through mid-office services, optimizing the maintenance of assets across the ecosystem.

The control office components pillar includes a partition that is growing to support a number of trends that all leverage new technology.

- The development of a componentized product portfolio (13) in which legal analysis tooling is used to create a new Regulation component (T),
- The introduction of the adaptable enterprise (14) through the new Business configuration component introduced in this paper (U),
- The deepening of business insights (15) using analytic technology to improve the existing Trend analysis component (V)

C.4 Commonality/Variability modelling

Fig. 12 illustrates the commonality and variability modelling approach. In the middle of Fig. 12 (the 3rd layer down from the top) three operating model components from our running example are shown: Contextual interactions, Single Window and Assessment. The bottom part of Fig. 12 (layer 4 and 5 down from the top) shows how for each operating model component a set of configurable features are embedded at a defined configuration point. We focus here on the modelling of the individual operating model components and how features are influenced by sources of variations. The overall design of features into an integrated space is discussed in more detail in the Engineering phase (see Appendix C.8). The top part of Fig. 12 (layers 1 and 2 down from the top) shows how selection of the appropriate features is governed by multiple sources of variations, tied to variation points.

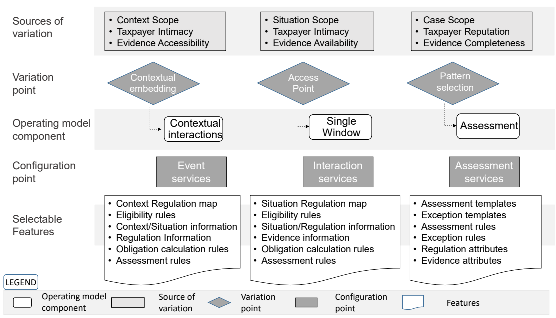

Figure 12: Operating model component variability modelling

The *Contextual Interactions* component integrates regulation specific information collection into an external context. For example, levying property tax on a real estate website. Selectable features that can be configured include:

- The Context Regulation map and Eligibility rules that contain the map and the decision rules that determine which regulations apply to the specific context,
- The Situation and Context information that is available through third parties (collected separately using the logistics pipelines),
- The Assessment and Obligation Calculation rules that are used for the assessment of the acquired data and for the calculation of the tax obligations.

Variations in the contextual embedding determine the scope of features. These include the context scope (influencing the scope of regulations that apply), taxpayer intimacy (influencing the eligibility of regulations for this taxpayer from this context, and if direct assessment and calculation of obligations is allowed), and accessibility of information (influencing the amount of data that needs to be collected directly from the context, and if direct assessment and calculation of tax obligations is allowed).

The *Single Window* component provides single access across all regulations that apply to a taxpayer's situation. Selectable features that can be configured include:

- The Situation Regulation map and Eligibility rules that contain the map and decision rules that determine which regulations apply to the specific taxpayer situation,
- The Situation information that needs to be collected from the taxpayer across regulations,
- The Regulation information that is available through 3rd parties (collected separately using the logistics pipelines),
- The Assessment and Obligation Calculation rules that determine the required assessment of data available and the tax calculation.

Sources of variations include the situation scope (influencing the scope of regulations that apply), taxpayer intimacy (influencing if direct assessment and calculation of obligations is allowed using the Single Window as a channel), and accessibility of evidence collected through 3rd parties (influencing the amount of data that needs to be collected directly from the taxpayer, and if direct assessment and calculation of obligations is allowed using the Single Window as a channel).

The *Assessment component* performs assessment of the correctness of a tax declaration. Selectable features that can be configured include:

- The Assessment and Exception templates that need to be performed, that structure data and processing,
- The Assessment and Exception rules to be applied that determine correctness of the declaration and if an exception needs to be raised that results in closer inspection,
- The Regulation and Evidence data available that are input for assessment and exception decisions.

Sources of variations include the case scope (influencing the templates and rules that apply), taxpayer reputation (influencing the exceptions that apply) and evidence completeness (influencing the amount of information that still needs to be validated through inspections).

C.5 Bandwidth modelling

The drivers that govern the trade-offs required for optimum bandwidth modelling are various. In the back office, operational excellence is the primary measure of outcome that drives choices. This puts a lot of focus on non-functional requirements such as robustness, throughput, etc. Operating model components are constructed with narrow bandwidths that are optimized for specific types of tax regulations and specific assessments. This produces multiple high-volume production lines that each cover a specific scope of regulations – for example to process simple filings for income tax assessments of taxpayers with good reputation.

On the other hand, in the front office – the Single Window component has been constructed with a scope that supports citizens in the widest possible sense. Customer intimacy is the driver in terms of outcome. This implies that a broad set of interaction styles is implemented (extensive guided dialogues, quick forms, chat facilities, help-desks, etc.) and that the interactions with the citizens are designed to match the context of their situation rather than the information needs of the individual tax product. To achieve a doable implementation with this broad scope, the Single Window utilizes integration junctions in the technology model to create an umbrella across individual back office operating model components that are more focused on a range of tax regulations. The plan is to eventually replace the usage of these junctions with a new front office operating model component with a broader scope that is situation focused. It will exploit new adaptability providing technology in the technology model to support this broader scope.

C.6 Family modelling

Family modelling at the layer of the business model for the tax agency implies the creation of families of regulation products. Regulations that the agency needs to enforce are split at the highest level in two separate groups of families – one for enterprises, one for citizens. Within these groups, separate families can be created for taxing of income (such as salaries, interest, shares, properties) and for compensations of cost of living (such as renting, child care, health insurance). Compare this structure with the example of a car manufacturer that sells both trucks and compact car, and that has several compact car families defined, such as vans, small town vehicles, and standard four-person cars.

Selectable features differ per family type and depend on the nature of the regulation. For example, the household composition for regulations that compensate for child related costs, or income data for income taxes, etc. Family hierarchies can be defined across all constituents of the business model. For example, channels split into two separate families: physical office channel and online website channel, with additional refinements at lower levels of the hierarchy.

Examples of feature modelling of operating model components are already included in Appendix C.4. To configure the processing of the different families of regulation products on the Single Window for example, features need to be defined that allow configuring the collection of situation information, the collection of evidence information, the determination of eligibility rules and the establishment of the interaction approach.

C.7 Business Configuration design

Tab. 4 includes a full elaboration of the contents of the BCC for the Single Window component for the different types of impact of the spectrum of change options. The first part of the table lists routine changes that can be handled by configuration within the bandwidth of existing operating model components), the second part requires adding new/extended operating model components, and the 3rd part requires adding full new operating model components in a new partition. Every row lists for a certain business variation the required change on the B2IT stack in the remaining columns: the business design, operating model and technology model. For example, the first row: if a new regulation is added to a family (for example 'vacation house' taxing) then the product catalog and targeting of the taxpayers needs to be extended (2nd column). This is used by the Establish Applicable regulation function to determine if a taxpayer is to be considered for this new tax. The features of the operating model component that need to be configured include the Situation Regulation map (that maps attributes of the taxpayer situation that need to be considered for this tax) and the Eligibility rules (that assess if the tax is applicable). Technology model components to which this metadata needs to be propagated include a Taxonomy handler (managing the Situation Regulation map) and a Rules handler (managing the Eligibility rules).

The basic metaphor that underlies this model is that of selling a family of configurable products/services with features that can be targeted (matched) to specific customer situations. For example, the family of property taxes, income taxes, child allowances, etc. Configurable features include product catalogues that list the regulations supported for different types of situations, eligibility rules that determine which regulations apply given this taxpayer's specific situation, regulation calculation rules that establish the tax obligations, questionnaires that collect taxpayer data and profile parameters that establish taxpayer characteristics. Feature extensions are required to support new functions that are added to these families such as processing topological information, contextualization, and delegation. Full new partitions are needed to support the new paradigm of delegated compliance processing that include setting up agreements and performing audits. Technology model components that provide adaptability in this example are middleware components. They center on the eligibility decisions, information processing, and calculations that the entry into an administrative agency requires. They include information handlers, rules handlers, and taxonomy and profile handlers, etc.

C.8 Integrated design of configuration

The example in Fig. 8 includes operating model components related to the use case that covers creation and processing of tax statements and evidence data. A tax statement can be created by accessing the Single Window from within a

Variation in Business that needs support	*Business Model Entity - Feature*	*Operating model component - Business Function*	*Operating model component - Feature*	*Technology model component*
Routine changes (can be handled by configuring within existing bandwidths)				
New/updated regulation within family	Product catalog, Product targeting	Establishing applicable regulation	Situation regulation map, Eligibility rules	Taxonomy handler, Rules handler
Yearly changes in taxation rules requiring additional questions, attributes, rules	Feature questionnaire	Collecting structured information	Regulation data attributes, Situation data attributes	Interaction handler, Data handler
Yearly changes in taxation rules requiring additional documents	Dropbox folder	Collecting unstructured information	Documents required	Document handler, Rules handler
Yearly changes in taxation segments requiring additional parameters	Customer profiling	Predefined customer characteristics	Profile parameters	Profile handler
Yearly changes in taxation rules requiring additional calculation rules, Yearly changes in processing regimes requiring new identification rules	Indicative offer and delivery date	Tax Handling	Assessment rules, Obligation calculation rules, Delivery indicator rules	Rules handler
Transitional changes (require extensions within existing landscape)				
Extension of family with new topographic features	Geo Map	Collecting topographic information	Selectable regions	Geo information handler
Extension of family with prefilled features	Information broker	Prefilled information	Evidence data attributes	Information handler
Extension of family with audiences that are supported by partners	Resellers	Delegated Handling	Authorization profiles	(Security) Profile handler
Extension of family with access from within ecosystem context requiring a new set of context features and selection of applicable regulations	Ecosystem embedded products	Contextualization	Context regulation map, Context features, Eligibility rules	Information handler, Rules handler, Mediation handler
Foundational changes (require new parts on top of existing landscape)				
New family of audit-based regulations requiring agreements types with configurable features	Contracts	Agreements framework	List of agreements, Agreement features	Taxonomy handler, Information handler
New family of audit-based regulations requiring a spectrum of audit regimes	Audit regimes	Performing audits of agreements	List of audit regime types, Audit features,Audit partners, Applicability rules	Taxonomy handler, Information handler, Profile handler, Rules handler

Table 4: BCC design - example

Context. It can be assessed and calculated if independent evidence has been obtained through the Logistics pipelines or it can be handed off for back office Assessment. Rules and data needs are derived from Regulation Development.

When adding a new tax regulation, all these operating model components will need to be extended with additional feature attributes or full new features. Integrated design is required across the feature space to ensure that the features are consistent across operating model components. Information features for example include contextual features and situation features, from which regulation specific data can be extracted. They can be provided directly from the context or by the taxpayer or be received as independent 3rd party evidence data through the logistics channel. A consistent design is required that aligns these features with potential different scopes across operating model components. The same applies for eligibility rules, assessment rules and calculation rules that depend on this information. They have potentially different scopes and different outcomes but need to be designed consistently across operating model components.

References

Abraham R., Tribolet J., Winter R. (2013) Transformation of Multi–Level Systems – Theoretical Grounding and Consequences for EnterpriseArchitectureManagement. In: Proper H., Aveiro D., Gaaloul K. (eds.) Advances in Enterprise Engineering VII. EEWC 2013. Lecture Notes in Business Information Processing Vol. 146. Springer, pp. 73–87

Al.Busaidi S., Kraiem N. (2017) Using Software Product Line Application in Enterprise Resources Planning Systems Systematic Literature Review. In: Journal of Computer Engineering Information Technology 6(3)

Alvertis I., Kokkinakos P., Koussouris S., Lampathaki F., Psarras J., Viscusi G., Tucci C. (2015) Challenges Laying ahead for Future Digital Enterprises: A Research Perspective. In: Persson A., Stirna J. (eds.) Advanced Information Systems Engineering Workshops. CAiSE 2015. Lecture Notes in Business Information Processing Vol. 215. Springer, pp. 195–206

Amoui M., Manesh M., Ebert J., Tahvildari L. (2012) Achieving Dynamic Adaptation via Management and Interpretation of Runtime Models. In: Journal of Systems and Software 85(12), pp. 2720–2737

Aparicio S., Aparício J., Carlos C. (2019) Data Science and AI: Trend Analysis. In: 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, pp. 1–6

Arsanjani A., Ghosh S., Allam A., Abdollah T., Ganapathy S., Holley K. (2008) SOMA: A Method for Developing Service-oriented Solutions. In: IBM Systems Journal 47(3), pp. 377–396

Ayora C., Torres V., Weber B., Reichert M., Vicente P. (2015) VIVACE: A Framework for the Systematic Evaluation of Variability Support in Process-aware Information Systems. In: Information and Software Technology 57(1), pp. 248–276

Baird C., Parasnis G. (2011) From Social Media to social Customer Relationship Management. In: Strategy Leadership 39(5), pp. 30–37

Barros A., Dumas M. (2006) The Rise of Web Service Ecosystems. In: IT Professional Magazine 8(5), pp. 31–37

Basolea R., Park H., Barnett B. (2015) Coopetition and Convergence in the ICT Ecosystem. In: Telecommunications Policy 39(7), pp. 537–552

Bengtsson P., Lassing N., Bosch J., van Vliet H. (2004) Architecture–level Modifiability Analysis (ALMA). In: Journal of Systems and Software 69(1–2), pp. 129–147

Bērziša S., Bravos G., Gonzalez T., Czubayko U., España S., Grabis J., Henkel M., Jokste L., Kampars J., Koc H., Kuhr J.-C., Llorca C., Loucopoulos P., Juanes R., Pastor O., Sandkuhl K., Simic H., Stirna J., Zdravkovic J. (2015) Capability Driven Development: An Approach to Designing Digital Enterprises. In: Business Information Systems Engineering 57(02), pp. 15–25

Bharadwaj A., Sawy O., Pavlou P., Venkatraman N. (2013) Digital Business Strategy: toward a Next Generation of Insights. In: MIS Quarterly 37(2), pp. 471–482

Blaga A., Tamas L. (2018) Augmented Reality for Digital Manufacturing. In: 26th Mediterranean Conference on Control and Automation (MED), Zadar, pp. 173–178

Blair G., Bencomo N., France R. (2004) Models@run.time. In: Computer 42(10), pp. 22–27

Blecker T., Friedrich G. (2006) Mass Customization: Challenges and Solutions Vol. 87. Springer

de Boer F., Bonsangue M., Groenewegen L., Stam A., Stevens S., van der Torre L. (2005) Change Impact Analysis of Enterprise architectures. In: Zhang D., Khoshgoftaar T. M., Shyu M.-L. (eds.) IRI -2005 IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, USA. IEEE, pp. 177–181

Bonini M., Urru A., Steinau S., Ceylan S., Lutz M., Schuhmacher J., Andrews K., Halfar H., Kunaschk S., Haque A., Nair V., Rollenhagen M., Shaik N., Reichert M., Bartneck N., Schlegel C., Hummel V., Echelmeyer W. (2018) Automation of Intralogistic Processes through Flexibilisation: a Method for the Flexible Configuration and Evaluation of Systems of Systems. In: ICINCO 2018: proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics: Porto, Portugal, July 29-31, 2018. - Volume 2. SCITEPRESS, Setubal, Portugal, pp. 380–388

Botta A., de Donato W., Persico V., Pescapé A. (2016) Integration of Cloud computing and Internet of Things: A Survey. In: Future Generation Computer Systems 56(3), pp. 684–700

vom Brocke J. (2007) Design Principles for Reference Models. Reusing Information Models by Aggregation, Specialisation, Instantiation, and Analogy. In: Fettke P., Loos P. (eds.) Reference Modeling for Business Systems Analysis. IGI Global, pp. 47–75

Bruls W., Baida Z., Koldijk F., Budel R., Meerman R., Kouwenhoven N. (2010) Information Management as a Sustainable Capability. In: Logistics Yearbook 2010: Your Guide to the World of Logistics and Supply Chain. Europse Vervoersorganisatie, pp. 93–103

Bruls W., Giesen E., Slaets H. (2016) Business Model Variability based Configuration Artefacts https://priorart.ip.com/IPCOM/000246781 Last Access: 2021-01-18

Bryan J., Kavantzas N. (2017) Defining Configurable Characteristics of a Product and Associating Configuration with Enterprise Resources. US Patent US 9,632,764 B2, 25 April 2017

Budiardjo E., Zamzami E. (2014) Feature Modeling and Variability: Modeling Syntactic Notation Comparison and Mapping. In: Journal of Computer and Communications 2(2), pp. 101–108

Caas (2021) Capability as a Service for Digital Enterprises. FP 7 ICT Programme Collaborative Project no: 611351. http://caas-project.eu

Campbell A., Gutierrez M., Lancelott M. (2017) Operating Model Canvas. Van Haren

Capilla R., Bosch J., Trinidad P., Ruiz-Cortés A., Hinchey M. (2014) An Overview of Dynamic Software Product Line Architectures and Techniques: Observations from Research and Industry. In: Journal of Systems and Software 91(3), pp. 3–23

Chau M., Ball G., Huang J., Chen J., Zhao L. (2011) Global IT and IT–enabled Services. In: Information Systems Frontiers 13(3), pp. 301–304

Clark T. (2018) Introduction to the Special Issue on Model-Driven Organisations. In: Enterprise Modelling and Information Systems Architectures (EMISAJ) – International Journal of Conceptual Modeling 13(3), pp. 1–3

Cognini R., Corradini F., Polini A., Re B. (2015) Extending Feature Models to Express Variability in Business Process Models. In: Persson A., Stirna J. (eds.) Advanced Information Systems Engineering Workshops. CAiSE 2015. Lecture Notes in Business Information Processing Vol. 215. Springer, pp. 245–256

Cusumano M. (2010) Cloud computing and SaaS as new computing Platforms. In: Communications of the ACM 53(4), pp. 27–29

Darking M. (2007) Understanding the Role of Governance in the Context of Digital Ecosystems. European Commission

De Haes S., Van Grembergen W. (2009) Enterprise Governance of Information Technology Achieving Strategic Alignment and Value. Springer

Demirkan H., Spohrer J. (2016) Emerging Service Orientations and Transformations (SOT). In: Information Systems Frontiers 18(3), pp. 407–411

Devadoss P. (2014) Deconstructing Enterprise Systems: Emerging Trends in the Next Generation of ES Applications. In: Topi H., Tucker A. (eds.) Computing Handbook Vol. 2. Chapman Hall

eCustoms (2021) An Industry Solution for Customs Administration.. IBM Global Business Services. https://www.ibm.com/downloads/cas/RBMYNVLX

Eldrandaly K., Abdel-Basset M., Shawky L. (2019) Internet of Spatial Things: A New Reference Model with Insight Analysis. In: IEEE Access 7, pp. 19653–19669

Engel A., Browning T., Reich Y. (2017) Designing Products for Adaptability: Insights from Four Industrial Cases. In: Decision Sciences 48(5), pp. 875–917

Engel A., Reich Y. (2015) Advancing Architecture Options Theory: Six Industrial Case Studies. In: Systems Engineering 18(4), pp. 396–414

España S., Grabis J., Henkel M., Koç H., Sandkuhl K., Stirna J., Zdravkovic J. (2015) Strategies for Capability Modelling: Analysis Based on Initial Experiences. In: Persson A., Stirna J. (eds.) Advanced Information Systems Engineering Workshops. CAiSE 2015. Lecture Notes in Business Information Processing Vol. 215. Springer, pp. 40–52

Evans N. (2017) Mastering Digital Business: How Powerful Combinations of Disruptive Technologies are Enabling the Next Wave of Digital Transformation Vol. 87. BCS

Evernden R. (1996) The Information Framework. In: IBM Systems Journal 35(1), pp. 37–68

Flaxer D., Nigam A., Vergo J. (2005) Using Component Business Modeling to Facilitate Business Enterprise Architecture and Business Services at the US Department of Defense. In: Proceedings of the 2005 IEEE International Conference on e-Business Engineering, Beijing, China, pp. 755–760

Fung C., Hung P. (2013) Information and Knowledge Management in online rich presence Services. In: Information Systems Frontiers 15(4), pp. 521–523

Galster M., Avgeriou P. (2013) Variability in Web Services. In: Capilla R., Bosch J., Kyo-Chul K. (eds.) Systems and Software Variability Management. Springer, pp. 269–278

Ganczarski W., Winter R. (2015) On the Interplay of Organizational Architecture and Software Architecture. In: Enterprise Modelling and Information Systems Architectures 3(1), pp. 24–35

Gerostathopoulos I., Bures T., Hnetynka P., Hujecek A., Plasil F., Dominik S. (2017) Strenghtening Adaptation in Cyber–physical Systems via Meta–adaptation Strategies. In: ACM Transactions on Cyber-Physical Systems 1(3), pp. 1–25

Giesen E., Berman S., Bell R., Blitz A. (2007) Three Ways to successfully Innovate your Business Model. In: Strategy Leadership 35(6), pp. 27–33

Girod S., Whittington R. (2019) Change Escalation Processes and Complex Adaptive Systems: From Incremental Reconfigurations to Discontinuous Restructuring. In: Organization Science 26(5), pp. 1520–1535

Goldkuhl G. (2004) Design Theories in Information: Systems – a Need for Multi-Grounding. In: Journal of Information Technology Theory and Application (JITTA) 6(2), pp. 59–72

Haftor D. (2015) Some Heuristics for Digital Business Model Configuration. In: Persson A., Stirna J. (eds.) Advanced Information Systems Engineering Workshops. CAiSE 2015. Lecture Notes in Business Information Processing Vol. 215. Springer, pp. 123–130

Hallsteinsen S., Stav E., Solberg A., Floch J. (2006) Using Product Line Techniques to Build Adaptive Systems. In: Proceedings - 10th International Software Product Line Conference, SPLC 2006, Baltimore, Maryland. IEEE, pp. 141–150

Harishankar R., Daley S. (2011) Actionable Business Architecture. In: 7th IEEE International Conference on E-Commerce Technology (CEC'05). IEEE Computer Society, pp. 318–324

Haugen Ø. (2013) Domain-Specific Languages and Standardization: Friends or Foes? In: Reinhartz-Berger I., Sturm A., Clark T., Cohen S., Bettin J. (eds.) Domain Engineering: Product Lines, Languages, and Conceptual Models. Springer, pp. 159–186

Haugen Ø., Møller-Pedersen B., Oldevik J., Olsen G., Svendse A. (2008) Adding Standardized Variability to Domain-Specific Languages. In: 12th International Software Product Line Conference, Limerick. IEEE, pp. 139–148

Henderson J., Venkatraman N. (1993) Strategic Alignment: Leveraging Information Technology for Transforming Orgnanisations. In: IBM Systems Journal 32(1), pp. 4–16

Hevner A. R., March S. T., Park J., Ram S. (2004) Design Science in Information Systems Research. In: MIS Quarterly 28(1), pp. 75–105

Hinkelmann K., Gerber A., Karagiannis D., Thoenssen B., van der Merwe A., Woitsch R. (2016) A new Paradigm for the Continuous Alignment of Business and IT: Combining Enterprise Architecture Modelling and Enterprise Ontology. In: Computers in Industry 79, pp. 77–86

Holl G., Grünbacher P., Rabiser R. (2012) A Systematic Review and an Expert Survey on Capabilities Supporting Multi Product Lines. In: Information and Software Technology 54(8), pp. 828–852

Horcas J., Pinto M., Fuentes L. (2017) Extending the Common Variability Language (CVL) Engine: A practical Tool. In: ter Beek M., Cazzola W., Diaz O., La Rosa M., López-Herrejón R., Thüm T., Troya J., Ruiz-Cortés A., Benavides D. (eds.) Proceedings of the 21st International Systems and Software Product Line Conference - Volume B (SPLC '17). ACM, pp. 32–37

Jansen S., Cusumano M., Brinkkemper S. (2013) Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry. Edward Elgar Publishing

Kang K., Cohen S., Hess J., Nowak W., Peterson S. (1990) Feature–Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-021. Carnegie Mellon University Software Engineering. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=11231

Keddis N., Kainz G., Zoitl A. (2015) Product-Driven Generation of Action Sequences for Adaptable Manufacturing Systems. In: IFAC-PapersOnLine 48(3), pp. 1502–1508

Kelly J., Hamm S. (2013) Smart Machines: IBM's Watson and the Era of Cognitive computing. Columbia University Press

Kephart J., Chess D. (2003) The Vision of Autonomic Computing. In: Computer 36(1), pp. 41–50

Klinger S., Beckner M. (2012) Formal Modelling of Components and Dependencies for Configuring Product-Service-Systems. In: Enterprise Modelling and Information Systems Architectures Journal 7(1), pp. 44–46

La Rosa M., van der Aalst W., Dumas M., Milani F. (2013) Business Process Variability Modelling: A Survey. In: ACM Computing Surveys 50(1), 2:1–2:45

Lankhorst M. (2012) Agile Service Development – Combining Adaptive Methods and Flexible Solutions. Springer

Lankhorst M. (2017) Enterprise Architecture at Work. Springer

Levi K., Arsanjani A. (2002) Mapping a Business Architecture to a Component–based Software Architecture. A Goal–drivenApproach to Enterprise Component Identification and Specification. In: Communications of the ACM 45(10), pp. 45–52

Li X., Eckert M., Martinez J.-F. a. R. G. (2015) Review Context Aware Middleware Architectures: Survey and Challenges. In: Sensors 15(8), pp. 20570–20607

Lim E., Chen H., Chen G. (2013) Business Intelligence and Analytics: Research Directions. In: Journal ACM Transactions on Management Information Systems 3(4), pp. 1–10

Maedche A., Gregor S., Morana S., Feine J. (2019) Conceptualization of the Problem Space in Design Science Research. In: Tulu B., Djamasbi S., Leroy G. (eds.) Extending the Boundaries of Design Science Theory and Practice. Lecture Notes in Computer Science Vol. 11491. Springer, pp. 18–31

Mannaert H., Verelst J., Ven K. (2012) Towards Evolvable Software Architectures based on Systems THeoretic Stability. In: Software Practice and Experience 42(1), pp. 89–116

March S., Smith G. (1995) Design and Natural Science Research on Information Technology. In: Decision Support Systems 15(4), pp. 251–266

Marshall A., Custis K., Lubowe D., White K. (2014) Individual Enterprise. GBE03614-USEN0-05. IBM Institute for Business Value. https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=XB&infotype=PM&appname=GBSE_GB_TI_USEN&htmlfid=GBE03614USEN&attachment=GBE03614USEN.PDF

Mazo R., Assar S., Salinesi C., Noura B. H. (Dec. 2014) Using Software Product Line to improve ERP Engineering: Literature Review and Analysis. In: Latin-American Journal of Computing 1(1)

Meertens L., Iacob M., Nieuwenhuis L., Jonkers H., van Sinderen M., Quartel D. (2012) Mapping the Business Model Canvas to ArchiMate. In: Ossowski S., Lecca P. (eds.) Proceedings of the 27th Annual ACM Symposium on Applied Computing, New York. ACM, pp. 1694–1701

Mistrík I., Ali N., Kazman R., Grundy J., Schmerl B. (2017) Managing Trade-offs in Adaptable Software Architectures. Morgan Kaufmann

Mongiello M., Di Noia T., Nocera F., Di Sciascio E. (2016) Case-based Reasoning and Knowledge-graph based Metamodel for Runtime Adaptive Architectural Modelling. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing, SAC 2016, New York. ACM, pp. 1323–1328

Morrisa K., Allison M., Costac F., Weid J., Peter C. (2015) An Adaptive Middleware Design to Support the Dynamic Interpretation of Domain–specific Models. In: Information and Software Technology 61(1), pp. 21–41

MSP (2021) Managing Successful Programs https://www.axelos.com/best-practice-solutions/msp/what-is-msp Last Access: 2021-01-18

Onix M., Fielt E., Gable G. (2017) Complex Adaptive Systems Theory in Information Systems Research – A Systematic Literature Review. In: PACIS 2017 Proceedings. AIS, pp. 1–14

Osterwalder A., Pigneur Y., Tucci C. (2005) Clarifying Business Models: Origins Present and Future of the Concept. In: Communications of the Association for Information Systems 16, pp. 1–25

Pastor O., Ruiz M., Koç H., Valverde F. (2018) Capability-based Communication Analysis for Enterprise Modelling. In: Enterprise Modelling and Information Systems Architectures (EMISAJ) 13(4), pp. 1–24

Peffers K., Tuunanen T., Rothenberger M. A., Chatterjee S. (2007) A Design Science Research Methodology for Information Systems Research. In: Journal of Management Information Systems 24(3), pp. 45–77

Persson A., Stirna J. (eds.) Advanced Information Systems Engineering Workshops. CAiSE 2015. Lecture Notes in Business Information Processing Vol. 215. Springer

Pichagonakesit T., Ueasangkomsate P., Sudharatna Y. (2019) A Review of Trends in Sustainable Manufacturing. In: 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON), pp. 202–205

Pohl K., Böckle G., van der Linden F. (2005) Software Product Line Engineering. Springer

Pohle G., Korsten P., Ramamurthy S. (2005) Component Business Models – Making Specialization real. G510-6163-00. IBM Institute for Business Value. https://www.ibm.com/downloads/cas/8RB5RWVE

Rabiser R., Wölfinger R., Grünbacher P. (2009) Three-Level Customization of Software Products Using a Product Line Approach. In: 2009 42nd Hawaii International Conference on System Sciences, pp. 1–10

Reinhartz-Berger I., Figl K. (2014) Comprehensibility of Orthogonal Variability Modeling Languages: the case of CVL and OVM. In: Gnesi S., Heymans P., Rubin J., Czarnecki K., Dhungana D., Fantechi A. (eds.) Proceedings of the 18th International Software Product Line Conference - Volume 1 (SPLC ’14). ACM, pp. 42–51

Reinhartz-Berger I., Figl K., Haugen Ø. (2017) Investigating Styles in Variability: Modeling: Hierarchical versus Constrained Styles. In: Journal Information and Software Technology 87(C), pp. 81–102

Revetria R., Tonelli F., Damiani L., Demartini M., Bisio F., Peruzzo N. (2019) A Real-Time Mechanical Structures Monitoring System based on Digital Twin, IoT and Augmented Reality. In: Spring Simulation Conference (SpringSim), Tucson, AZ, USA, pp. 1–10

Rodrìguez G., Soria A., Campo M. (2016) Artifical Intelligence in Service-oriented Software Design. In: Software Systems Modeling 53, pp. 86–104

Romeroa D., Vernadat F. (2016) Enterprise Information Systems (state) of the Art: Past Present and Future Trends. In: Information Systems in Management 79, pp. 3–13

Ross J., Weill P., David R. (2006) Enterprise Architecture as Strategy. Harvard Business School Press

Rurua N., Eshuis R., Razavian M. (2017) Representing Variability in Enterprise Architecture. In: Business Information Systems Engineering 61(2), pp. 215–227

Schilling R., Beese J., Haki K., Aier S., Winter R. (2017) Revisiting the Impact of Information Systems Architecture Complexity: A Complex Adaptive Systems Perspective. In: Thirty Eighth International Conference on Information System (ICIS 2017)

Schmidt A. (2007) IT and the Judiciary in the Netherlands – A State of Affairs. In: Computer Law Security Review 23(12), pp. 453–460

Schnieders A. (2006) Variability Mechanism Centric Process Family Architectures. In: Proceedings of the 13th Annual IEEE International Symposium and Workshop on Engineering of Computer Based Systems, Potsdam, Germany, 27-30 March 2006. IEEE, pp. 141–150

Sessions R. (2008) Simple Architectures for Complex Enterprises. Microsoft Press

Al-Shardan M. M., Ziani D. (2015) Configuration as a Service in Multi-Tenant Enterprise Resource Planning System. In: Lecture Notes on Software Engineering 3(2), pp. 95–100

Simon H. (1996) The Science of the Artificial. MIT Press

Song M., Jung Y. J., Oh J., Choi S. (2015) Oracle: Stacking high through Technology Integration. In: Journal of the International Academy for Case Studies 21(3), pp. 223–241

Stoicescu M., Fabre J., Roy M. (2017) Architecting Resilient Computing Systems: A Component-based Approach for Adaptive Fault Tolerance. In: Journal of Systems Architecture 73, pp. 61–16

Strong D., Volkoff O. (2005) Understanding Organization–Enterprise System Fit: a Path to Theorizing the Infrmation Technology Artefact. In: MIS Quarterly 34(4), pp. 731–775

Szvetits M., Zdun U. (2016) Systematic Literature Review of the Objective Techniques Kinds and Architectures of Models at Runtime. In: Software Systems Modeling 15(1), pp. 31–69

Tanriverdi H., Rai A., Venkatraman N. (2010) Research Commentary – Reframing the dominant Quest of Information Systems Strategy Research for Complex Adaptive Business Systems. In: Information Systems Research 21(4), pp. 822–834

Thuan N. H., Drechsler A., Antunes P. (2019) Construction of Design Science Research Questions. In: Communications of the Association for Information Systems 44

TOGAF (2021) The Open Group Architecture Framework http://www.opengroup.org/togaf Last Access: 2021-01-18

van der Torre L., Lankhorst M., ter Doest H., Campschroer J., Arbab F. (2013) Landscape Maps for EnterpriseArchitectures. In: Duboi E., Pohl K. (eds.) Advances in Enterprise Engineering VII. EEWC 2013. Lecture Notes in Business Information Processing Vol. 4001. Springer, pp. 351–366

Van Landuyt D., Walraven S., Joosen W. (2015) Variability Middleware for Multi-Tenant SaaS Applications: A Research Roadmap for Service Lines. In: SPLC '15: Proceedings of the 19th International Conference on Software Product Line. ACM, pp. 211–215

Velicogna M. (2007) Justice systems and ICT: What can be learned from Europe? In: Utrecht Law Review 3(1), pp. 129–147

Wang L., von Laszewskim G., Younge A., He X., Kunze M., Tao J., Fu C. (2010) Cloud computing: a Perspective Study. In: New Generation Computing 28(2), pp. 137–146

Wareham J., Fox P., Giner J. (2014) Technology Ecosystem Governance. In: Organization Science 25(4), pp. 1195–1215

Wasilewski A. (2015) Business process management suite (BPMS) Market changes 2009-2015. In: Information Systems in Management 5(4), pp. 585–592

Wehling K., Wille D., Pluchator M., Schaefer I. (2016) Towards Reducing the Complexity of Enterprise Architectures by Identifying Standard Variants Using Variability Mining. In: 1. Automobil Symposium Wildau: Tagungsband, pp. 37–43

Weyns D. (2019) Software Engineering of Self-adaptive Systems: An Organised Tour and Future Challenges. In: Cha S., Taylor R. N., Kang K. C. (eds.) Handbook of Software Engineering. Springer, pp. 399–433

Whitmore A., Agarwal A., Da Xu L. (2013) The Internet of Things – A Survey of Topics and Trends. In: Information Systems Frontiers 17(4), pp. 261–274

Winter R. (2011) Problem Analysis for Situational Artefact Construction in Information Systems. In: Carugati A., Rossignoli C. (eds.) Emerging Themes in Information Systems and Organization Studies Vol. 11491. Springer, pp. 97–113

Vol. 16, No. 5 (2021). DOI:10.18417/emisa.16.5

Zimmermann A., Gonen B., Schmidt R., El-Sheikh E., Bagui S., Wilde N. (2014) Adaptable Enterprise Architectures for Software Evolution of SmartLife Ecosystems. In: 18th International Enterprise Distributed Object Computing Conference Workshops and Demonstrations (EDOCW), Ulm, Germany, 1-2 Sept 2014. IEEE, pp. 316–323

Zimmermann A., Schmidt R., Jugel D., Möhring M. (2015) Evolving enterprise architectures for digital transformations. In: Zimmermann A., Rossmann A. (eds.) Digital Enterprise Computing (DEC 2015). Gesellschaft für Informatik e.V., pp. 183–194