BITA*: Business-IT Alignment Framework of Multiple Collaborating Organisations

Kassahun, A., & Tekinerdogan, B.

Kassahun, A., & Tekinerdogan, B. (2020). BITA*: Business-IT Alignment Framework of Multiple Collaborating Organisations. Information and Software Technology, 127, [106345]. https://doi.org/10.1016/j.infsof.2020.106345

BITA: Business-IT Alignment Framework of Multiple Collaborating Organisations*

Abstract

Context: Businesses today must collaborate in a coordinated fashion. To collaborate, they must align their business processes and IT by complying to a common reference architecture. The common reference architecture that addresses their specific collaboration requirements is generally an adaptation of an existing generic reference architecture. However, a design framework for adapting reference architectures is lacking.

Objective: In this paper we propose a design framework for aligning business processes and IT across diverse collaborating organisations in order to derive a more specific reference architecture from a generic one.

Method: We developed the design framework using the guidelines of ISO/IEC/IEEE standard for modelling design viewpoints and validated it in a real-life business case study.

Results: We developed an architectural design framework which we call BITA* that is composed of three coherent architectural design viewpoints. The BP2BP alignment viewpoint provides alignment modelling abstractions for business analysts to be used to align business collaboration processes. The IT2IT alignment viewpoint provides alignment modelling abstractions for software architects to be used to align distributed IT systems. The BP2IT alignment viewpoint provides alignment modelling abstractions for interdisciplinary teams of business and IT specialists for aligning the mapping of business collaboration processes and the underlying distributed IT. The modelling abstractions are applied in a case study to derive a reference architecture for meat supply chain transparency systems.

Conclusion: A key challenge in developing the design framework is the difficulty of comparing models of business processes and IT that come from diverse organisations. Our main contribution is the set of modelling abstractions, which enabled us to represent business processes and IT in a uniform and comparable manner, and the systematic approach for applying the modelling abstractions. The framework is applied in the agri-food sector and needs to be evaluated further in multiple case studies from various application domains.

Keywords: Business-IT alignment; business collaboration; reference architecture; distributed systems; business process models; workflow patterns

1 INTRODUCTION

To achieve their goals, businesses today rarely operate in isolation but must collaborate in a variety of processes with others in a coordinated fashion. To support the collaboration, their Business Processes (BPs) and the underlying Information Technology (IT) must be well-aligned. This implies that the BP and IT models of the different collaborating organisations should be made interoperable with each other to realize business integration.

In fact, business-IT alignment is not a new problem but has been broadly addressed in literature. However, the alignment problem has mainly been addressed within the context of a single organisation. When dealing with multiple organisations, the problem is not addressed consistently. Generally, one of the following two different approaches is used. The first approach focusses on the IT alignment problem and applies pairwise inter-organisational alignments through service orientation of the IT systems (Chen 2008, Demirkan *et al.* 2008, Aversano *et al.* 2016). The second approach focusses on alignment of BPs in order to make them executable across organisational boundaries (Peltz 2003, Newcomer and Lomow 2004, Erl 2008, Liu *et al.* 2009, Cummins 2015).

Alignment across multiple organisations requires that the BPs and IT systems needed for collaboration are supported and are compatible with each other. Some features of the required BPs and IT must be supported by all collaborating organisations, while some other features, or entire BPs and IT systems, need to be supplied by some of the collaborating organisations only, or even by third parties. Alignment not only ensures BPs and IT systems are interoperable but, and most importantly, it ensures they are supported by the right collaboration partners. For instance, tracking and tracing of food products in the agriculture and food (agri-food) sector requires all food operators to capture the required transparency data, but only some of the organisations, often a focal company (or a third party), need to provide IT systems that will aggregate and present transparency information to end users. Alignment in this case deals with identifying which organisations should support which business activities and IT systems. Thus, though the two approaches mentioned in the previous paragraph are required for alignment, the way the approaches are currently applied creates major drawbacks when the number of organisations involved increases substantially. First, though it is possible to align IT and BP models pairwise, such as approach to alignment involves *nC2* (n combination 2) possible alignments, where *n* is the number of organisations. This is probably only acceptable for a small number of organisations. Second, though executability of BPs can be modelled using orchestration and choreography languages, such an approach requires BPs and IT systems that are aligned in the first place and all components are supplied by the right party.

A strategy commonly adopted to address both of the above issues is to comply with a common *reference architecture*. When a reference architecture is adopted, only as many alignments are required as there are organisations. This means that instead of *nC2*, only *n* alignments are required. Reference architectures define reference models. These models can be reference BP models, references IT models or reference mappings of BP and IT (*i.e.* BP-IT mapping) models. Over the years a number of generic reference architectures have been defined by global standardisation bodies, such as BP modelling related standards by Object Management Group (OMG 2017), web-service (IT) related reference models by Organisation for the Advancement of Structured Information Standards (OASIS 2017), traceability standards by Global Standards 1 (GS1 2017), enterprise architecture reference model by The Open Group (The Open Group 2017), and Supply Chain Reference Model (SCOR) by and Supply Chain Council (SCC), which is now part of the American Production and Inventory Control Society (APICS 2017).

Though the available reference architectures are valuable, they are usually too generic to address the unique requirements of a specific set of collaborating organisations or a specific sector of an industry. Therefore, existing reference architectures are often adapted and extended in order to make them suitable for the specific application context. For instance, the SCOR and GS1 reference architectures are applicable for all sectors, but some sectors, such as the agri-food sector, have unique requirements due to their unique characteristics. The agri-food sector involves many small food operators, deals with perishable products, and those products sometimes involve many product transformations. Therefore, a number of agri-food specific reference architectures have in the past been developed based on the generic SOA (Service-Oriented Architecture), SCOR and GS1 reference architectures (Steinberger et al. 2009, Verdouw et al. 2010, Wolfert et al. 2010, Kruize et al. 2016).

There are however no design frameworks that guide the development of a new reference architecture from existing generic reference architectures. In this paper we propose BITA*, which is an approach for deriving a reference architecture. A reference architecture consists of aligned BP and IT models that enable collaboration across multiple organisations. In the approach adopted in BITA*, a new reference architecture is derived through a process of alignment of existing BP and IT models used by the collaborating organisations (concrete models) with the BP and IT models of suitable generic reference architectures (reference models). BITA* thus stands for BP-IT Alignment (BITA) framework and the symbol '*' denotes that multiple organisations are involved.

In order to understand the need for such a design framework, it is important to understand how the BPs and IT systems of collaborating organisations can be misaligned. Collaborating organisations generally define cross-organisational BPs, such as, planning, procurement and sales BPs. They also define the associated IT interfaces for sharing information. Though each organisation at some point designs and redesigns BP and IT models for the purpose of collaboration; the models are often misaligned when many organisations are involved. An example of large-scale collaboration can be found in the agri-food sector where many feed, breeding, fattening and meat processing businesses collaborate in complex supply chains. A commonly occurring misalignment is the lack of interoperable interfaces and data models for retrieving product provenance information. A reference architecture would address such alignment problems by defining reference interfaces and data models that each collaborating organisation should comply with.

We refer hereafter BP an IT models that are meant for collaboration purposes as *Business Collaboration Models* (*BCMs*). BCMs come in three variants. The first type of BCMs are BP models for collaboration, which we refer to as *Business Collaboration Process* (*BCP*) models. The second type of BCMs are IT models that represent the distribution and integration of the IT across the collaborating organisations, which we refer to as *Distributed IT* (*DIT*) models. A complete representation of DIT may require the use of an extensive set of DIT architectural patterns (Buschmann *et al.* 2007), which is beyond the scope of this study. We use in this study a limited aspect of DIT, mainly models for representing shared data objects and IT interfaces that are often referred to as web services. The last type of BCMs are models that represent the mapping of *BCPs* to the underlying *DIT*, which we refer to as *BCP-DIT* models. A BCP is typically modelled as a BPMN (ISO/IEC 2013) collaboration model. A DIT model can typically be a common data standard and web-service interface specification, and BCP-DIT model can be a mapping of a business activity of one organisation (identified in a BPMN model) to a data object and IT interface provided by another organisation (identified in data and IT service model).

We identify three ways in which BCMs can be misaligned. First, each individual organisation adopts a number of *BCP* models, such as procurement and sales BCP models. Ideally, these BCP models should be interoperable with each other—for instance, a procurement BCP of customers should be aligned with the corresponding sales BCP models of suppliers. Unfortunately, that is not always the case. In order to align their BCPs, organisation undertake a lengthy negotiation and alignment of their BCPs. We call this *BP to BP (BP2BP) alignment concern*. Second, each organisation has its own models on how to support BCPs by the underlying DIT (represented as BCP-DIT models), and these models may be misaligned. We call this *BP to IT (BP2IT) alignment concern*. Third, each organisation's models governing how data and IT systems should be distributed among the collaborating organisations to form a DIT also differ, and we call this *IT to IT (IT2IT) alignment concern*. The three different types of alignments that needs to be addresses are depicted graphically in Figure 1.

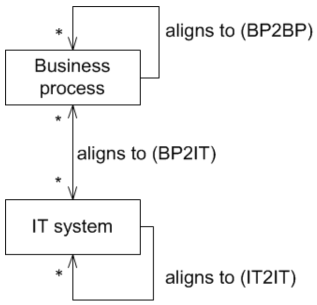

Figure 1: Business-IT alignment in multi-organisational collaboration.

The different alignment concerns are addressed differently in different literature. So far, a coherent set of explicit design abstractions and the corresponding design heuristics for aligning misaligned models across multiple collaborating organisations are largely missing. Business analysts and software architects lack the tools for deriving reference BCMs and as a result, the process of deriving a reference architecture from existing generic reference architectures remains ad hoc and informal. In BITA*, we provide both the required alignment modelling abstractions as well as a systematic approach that is necessary for applying the modelling abstractions. Alignment modelling is the core of the BITA* framework and is demonstrated using a real-life business case.

The rest of the paper is structured as follows. In section 2 we present a review of related literature. In section 3 we provide background information about the building blocks of the alignment framework. In section 4 we explain the research methodology and in section 5 we present the BITA* alignment design framework. To validate the framework, we applied it to a business case study in section 6. In section 7 we discuss the application of the approach and, finally, we make concluding remarks in section 8.

2 RELATED WORK

There is a considerable literature on the three types of alignment concerns though they are not presented as such. BP2BP alignment concerns have largely been addressed in management and business literature, such as business process outsourcing (Davenport 2005), business process compliance (Sadiq *et al.* 2007), and business process maturity (Rosemann and vom Brocke 2015). BP2IT alignment concerns have, in fact, gained much attention in the business-IT alignment literature but mainly within the limited scope of the individual business organisation (or enterprise). BP2IT alignment concerns have often been seen as a one-way design problem where the design of an IT system is considered to be guided by business process models (Zachman 1987, Wieringa *et al.* 2003, The Open Group 2011). BP2IT alignment concerns have so far not been addressed systematically as a two-way alignment problem. IT2IT alignment concerns have largely been addressed as coupling, integration or interoperability issues (Chen *et al.* 2008, Daclin *et al.* 2016).

BP-IT alignment has been addressed as such by Chen et al. (2005), who proposed BITAM (Business IT Alignment Method). BITAM is a methodology that consists twelve steps for detecting and correcting misalignments. Typical steps of their approach include: *elicit business and IT architecture from architects* (step 5 and 6), *map operational scenarios onto business and IT architectures* (steps 7 and 8), and *assess the misalignments* (step 9). However, the approach depends on personal perceptions and not on explicit models.

Recently, Hinkelmann et al. (2016) propose a business and IT alignment approach that combines enterprise architecture modelling (including the modelling approaches we used in this paper) and enterprise ontologies. Enterprise ontologies are tools of knowledge engineering and enable explicit specifications of conceptualizations of a given problem domain (Gruber 1995) and will enable building a knowledge base of explicit representation of reference models and patterns. However, the authors did not present a knowledge base that is comparable to the workflow patterns that we successfully applied.

Yet another recent related work focused on a specific aspect of misalignment between business processes and software user interfaces (Hoch *et al.* 2016). The authors identify the fact that gaps between business processes and IT exist because the representation of process elements in the IT models is implicit—a case in point being the lack of user interface specification in BPMN models. They have proposed a model of representing business artefacts to enrich BPMN models so that implicit assumptions of business process and the unforeseen business-IT misalignments can be avoided.

In addition to these and other business-IT alignment approaches, the existing enterprise designs frameworks, such as TOGAF (The Open Group 2011), provide methodologies that address alignment issues. However, these frameworks can be characterised as a one-way alignment methods because they guide how to design the IT to fulfil the requirements laid out in the form of business process models, but not the other way round. Generally, the common limitation of existing alignment approaches and enterprise design frameworks is that they do not provide abstractions for comparing models in an objective manner and do not address the specific alignment problems of multiple collaborating organisations.

3 BACKGROUND

The required building blocks of the business-IT alignment framework are BP models, workflow patterns and IT models. These are presented in sections 3.1, 3.2, and 3.3, respectively.

3.1 BUSINESS PROCESS MODELS

BP models are formal mechanisms for defining BPs. Originally introduced for modelling collaboration among functional departments of an organisation (Davenport and Short 1990, Hammer 1990, Harrington 1991), BP models are, nowadays, extensively used to model collaboration across organisational boundaries (Wolfert *et al.* 2010, Alotaibi 2016, Pradabwong 2017).

A recent survey suggests that the most widely adopted approach for modelling BPs is BPMN (Harmon 2016). BPMN stands for Business Process Model and Notation (ISO/IEC 2013) and is a formal specification for modelling BPs. BPMN provides three types of models: process model (Chapter 10, ISO/IEC 2013), collaboration model (Chapter 9, ISO/IEC 2013), and choreography model (Chapter 11, ISO/IEC 2013). A process model describes a BP by specifying the sequencing of business activities within a particular organisational unit. Collaboration and choreography models are used to model BCPs. A model of collaboration across organisational boundaries, in its simplest form, consists of pools across which messages are exchanged. A choreography model describes the interactions (instead of message exchanges) among the collaborating organisations. A choreography model is essentially a different form of representing a collaboration model.

To interoperate, organisations need to comply with a common set of generic BCPs. These BCPs are generally provided as reference models, of which SCOR (SCC 2012) and the BP models of the GS1 traceability standard (GS1 2017) are good examples.

3.2 WORKFLOW PATTERNS

BP modelling primarily focuses on how to represent the different process workflows. However, BP models generally contain many recurring patterns that BP modellers often come across. These recurring problem-solution pairs are called workflow patterns (Russell *et al.* 2006). In the past, more than a hundred workflow patterns have been identified, categorized and catalogued (van der Aalst and ter Hofstede 2011). The most prominent categories are *control-flow*, *data-flow* and *resource-flow* patterns (Van der Aalst *et al.* 2003). A Control-Flow Pattern (CFP) defines a recurring pattern of sequencing of activities in BP models. A Data Flow Pattern (DFP) models the patterns of data access and usage. Resource Flow Patterns (RFPs) define the patterns of resource allocations in BPs. In this paper we apply CFPs and DFPs only.

3.2.1 Control-Flow Patterns

Several CFPs have been defined and categorized by van der Aalst and ter Hofstede which are summarized in Table 1. We identify four categories of CFPs: *branch and sync*, *iteration*, *multiple instance* and *event-driven*. *Branch and sync* CFPs define the sequencing of activities, such as linear (sequential), branching and parallel. *Iteration* CFPs define how the same sets of activities are performed repetitively. Iteration can be a loop or a recursion. *Multiple instance* CFPs define how the same sequence of activities is executed in parallel in separate threads of execution. *Event driven* CFPs define the effects of expected and unexpected events, such as starting, cancelling and completion. CFPs can be arranged hierarchically (*i.e.* a CFP can not only contain activities but it can also contain other CFPs) and as such are a powerful means of capturing BP models at various levels of abstraction.

Table 1: Control-flow patterns (adapted from, van der Aalst and ter Hofstede 2011).

Pattern Categories	Patterns*
Branch and Sync	Sequence (1), Parallel Split (2), Synchronization (3), Exclusive Choice (4), Simple Merge (5), Multi-Choice (6), Structured Synchronizing Merge (7), Multi-Merge (8), Structured Discriminator (9), Blocking Discriminator (28), Cancelling Discriminator (29), Structured Partial Join (30), Blocking Partial Join (31), Cancelling Partial Join (32), Generalized AND-Join (33), Local Synchronizing Merge (37), General Synchronizing Merge (38), Thread Merge (41), Thread Split (42), Deferred Choice (16), Interleaved Parallel Routing (17), Milestone (18), Critical Section (39), Interleaved Routing (40)
Iteration	Arbitrary Cycles (10), Structured Loop (21), Recursion (22)
Multiple Instance	Multiple Instances without Synchronization (12), Multiple Instances with a Priori Design-Time Knowledge (13), Multiple Instances with a Priori Run-Time Knowledge (14), Multiple Instances without a Priori Run-Time Knowledge (15), Static Partial Join for Multiple Instances (34), Cancelling Partial Join for Multiple Instances (35), Dynamic Partial Join for Multiple Instances (36)
Event-driven	Transient Trigger (23), Persistent Trigger (24), Cancel Task (19), Cancel Case (20), Cancel Region (25), Cancel Multiple Instance Activity (26), Complete Multiple Instance Activity (27), Implicit Termination (11), Explicit Termination (43)

* The pattern names used by the authors are shortened for the sake of readability; the pattern IDs (given insides brackets) are, however, original.

3.2.2 Workflow data pattens

DFPs are used to capture well-known and recurring data flows. Table 2 lists the DFPs that are relevant for representing data sharing concerns in multi-organisational collaboration context. The patterns are categorized into four categories by van der Aalst and ter Hofstede, namely: *visibility*, *interaction*, *transfer* and *routing* DFPs. This categorisation is important because the data access and usage concerns fall also into these categories. *Visibility* DFPs define the scope of accessibility of a data object. For instance, an *activity*[1] *scope* signifies that the data object visibility is restricted to the activity instance; while an *instance scope* signifies that its visibility extends to all activities of a BP instance. *Interaction* DFPs define how the data object visibility changes due to interaction. For instance, *activity to activity* means that the data object remains in an activity scope during interaction; while *to multiple instance activity* means that the interaction changes to *multiple instance* scope. *Transfer* DFPs define the mechanisms of data interaction, which can be *by value*, *by reference*, etc. *Routing* DFPs define how a data object affects the control flow, such as launching or ending an activity, or altering the flow of control.

---

[1] We use the term *activity* instead of *task* (the term originally used in DFPs) in order to be consistent with the terminology of BPMN. We also use the term *activity* instead of *task* and *block task; instance* instead of *case*; *business process* instead of *workflow*, and (*external*) *data store* instead of *environment.*

Table 2: Workflow data patterns (adapted from, van der Aalst and ter Hofstede 2011).

Categories	Patterns*
Visibility	Activity (1), Multiple Instance (4), BP Instance (5), External (8)
Interaction	
Internal	Activity to Activity (9), To Multiple Instance Activity (12), From Multiple Instance Activity (13), Instance to Instance (14)
External	Activity pushes data (15), Activity pulls data (16), Data are pushed to Activity (17), Activity receives data (18), BP Instance pushes data (19), Data are pulled from BP Instance (20), Data are pushed to BP Instance (21), BP Instance pulls data (22)
Transfer	Incoming By Value (27), Outgoing by Value (28), Copy In/Copy Out (29), By Unlocked Reference (30), By Locked Reference (31), Input Transformation (32), Output Transformation (33)
Routing	Existence as Activity Precondition (34), Value as Activity Precondition (35), Existence as Activity Postcondition (36), Value as Activity Postcondition (37), Event-based Activity Trigger (38), Data-based Activity Trigger (39), Data-based Routing (40)

* The pattern names used by the authors are shorted for the sake of readability; the pattern IDs (given insides brackets) are, however, original. DFPs deemed irrelevant for the purpose of this paper are not included.

3.3 IT MODELS

The basic artefact that shapes the design of IT systems is software architecture. Software architecture defines the components of the software system of an organisation, the interactions among the components, and the interaction of the system as a whole with its environment (ISO/IEC/IEEE 2011).

Collaboration involves IT systems that are distributed across collaborating organisations (including nowadays IT service providers). Just like the integration of BPs, the integration of IT systems requires compliance with a common specification, generally referred to as a reference architecture. A reference architecture guides the design of the concrete architectures of the collaborating organisations (Cloutier *et al.* 2010, Angelov *et al.* 2012). Hereby, concrete architecture refers to a software architecture for a specific context (*i.e.* for a particular organisation or a set of organisations) and that which can be implemented into a software system.

A reference architecture for a distributed system is largely defined by a common data model and a set of IT service models. In SOA, an IT service represents an interface, generally a web-service interface, of an IT system or a BP that is exposed as an IT service. According to the SOA approach, collaborating organisations take one or more of the following roles: *service client*, *service provider* and *service broker* (OASIS 2006). The desired integration is then achieved when organisations *publish* their IT services at a third party *discovery* service so that their *clients* (collaborating partners) can find the IT services and use them to *exchange messages* based on standardised data and interface protocols (Barry 2003, Papazoglou *et al.* 2008, Buyya *et al.* 2009).

4 RESEARCH METHOD

This study follows a design science research methodology following Hevner *et al. (2004)*. According to Hevner *et al.*, design science research consists of *relevance, design* and *rigor* cycles, which is consistent with the approach followed in this study. The relevance cycle provides the justification for developing new design artefacts and identifies the requirements. In modern software development, requirements are gathered from stakeholders incrementally through various methods, such as brainstorming, domain analysis and prototyping (Nuseibeh and Easterbrook 2000, Laplante 2013). The requirement for an alignment framework came from two large-scale integration projects conducted in the agri-food sector, which are the SmartAgriFood and FIspace projects. Within these projects, requirements were gathered for the alignment and integration of agri-food information systems, and prototype systems were built based on those requirements (Kaloxylos *et al.* 2013, Verdouw *et al.* 2014, Barmpounakis *et al.* 2015). The challenges faced in those projects provided the basis for the alignment framework.

The design cycle creates new design artefacts for addressing the problem at hand (Simon 1996). Designs are created using an existing body of design knowledge (Hevner *et al.* 2004, Hevner and Chatterjee 2010). The existing body of knowledge for this research constitute BPMN (ISO/IEC 2013), workflow patterns (van der Aalst and ter Hofstede 2011) and SOA (OASIS 2006). We combined these modelling abstractions and created new modelling abstractions. For designing the framework, we applied the notion of design viewpoint that is widely used in the context of software architecture. A design *viewpoint* specifies the required modelling abstractions for addressing the specific concerns of specific type of stakeholders. Stakeholders use a viewpoint and follow its conventions, including the model types and notations, to create a design which is referred to as a *view* (ISO/IEC/IEEE 2011). Various examples of architecture viewpoints are provided in the literature (Clements *et al.* 2010). Our framework is a design framework, and in terms of Gregor and Hevner (2013) classifications of design science research, our contributions can be classified as a “level 2” nascent design theory.

The rigor cycle ensures that the resulting design is valid. We applied a case study research methodology for information systems research (Easterbrook *et al.* 2008, Runeson and Höst 2008) in the rigor cycle. The case study in this research comes from the FIspace project (FIspace 2013) conducted from 2013 to 2015.

5 BITA* FRAMEWORK

The following sub-sections present the elements of the BITA* framework. In section 5.1, we present the BITA* metamodels. In section 5.2 we describe the systematic approach for applying BITA*. Finally, in section 5.3, we present the modelling abstractions we developed grouped into three alignment viewpoints.

5.1 METAMODEL

According to the ISO/IEC/IEEE standard (ISO/IEC/IEEE 2011), a design framework consists of design viewpoints and each viewpoint addresses the concerns of a specific stakeholder type. Viewpoints provide model types with which the concerns of the stakeholders and the corresponding solutions can be expressed. The concerns generally related to the AS-IS situation, and the solutions represent the desired TO-BE situation. We introduced three distinct *alignment viewpoints*, which are *BP2BP*, *IT2IT, and BP2IT* viewpoints. The three viewpoints correspond to the three stakeholder types, *business analysts*, *software architects* and *interdisciplinary teams of business analysts and software architects*, respectively. Each alignment viewpoint consists of two model types: *allocation* and *alignment* model types. Which means that each of the tree viewpoints define specific types of allocation and alignment modelling abstractions. The high-level metamodel of the BITA* framework is depicted in Figure 2.

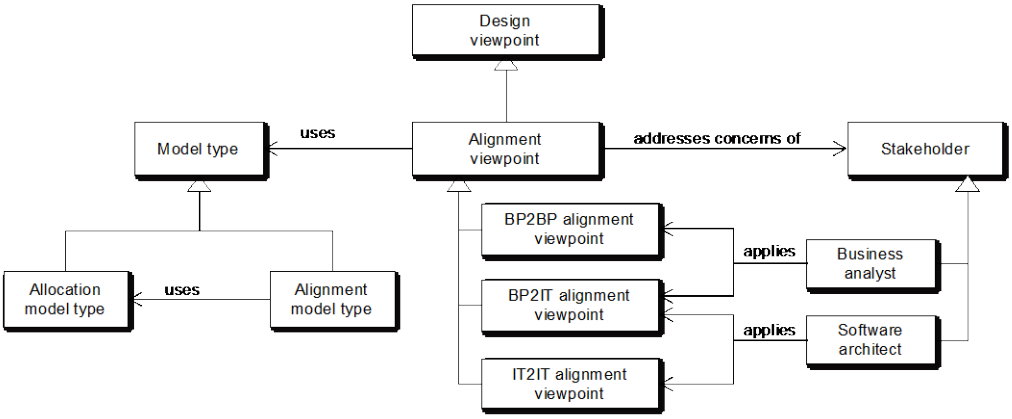

*Figure 2: The high-level metamodel of BITA**

The viewpoints of BITA* provide a simple and objective means of comparing existing (AS-IS) or current version of BCMs with the possible future (TO-BE) BCMs. A BCM can be a BCP, DIT, or BCP-DIT model as described in the introduction. BCMs are expressed using existing body of modelling knowledge, namely BCPs are expressed using BPMN collaboration (interaction) models, and DIT and BCP-DIT are expressed using SOA IT service models. SOA IT service models include many separate models, prominently IT service descriptions (W3C 2007), message (data) exchange protocols (Mitra 2003, Bouguettaya *et al.* 2014), and IT service discovery models (OASIS 2004, Crasso *et al.* 2013). This aspect of the BITA* metamodel is depicted in Figure 3.

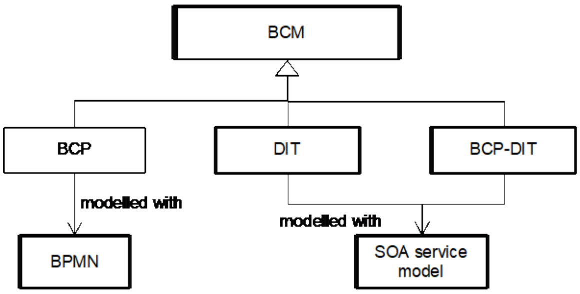

*Figure 3: Business collaboration modelling abstractions of BITA**

Core to BITA* are its modelling abstractions (model types in the terminology ISO/IEC/IEEE) that are associated with the alignment viewpoints. Specifically, we introduce the *allocation* and *alignment* modelling abstractions that build on the existing body of BP and IT modelling knowledge. BITA* provides specific allocation and alignment modelling abstractions in each viewpoint, which means that it provides unique alignment modelling abstractions for each stakeholder type.

Allocation and alignment models are introduced because a BCM (based on BPMN and SOA service models) cannot easily be compared with a different version of it (for instance, a concrete BCM with a reference BCM), and as a result it is difficult to make explicit statement about the state of the alignment. In order to make BCMs comparable, we provide a mechanism of converting BCMs into matrix-based representation with the help of workflow patterns, specifically DFPs and CFPs. Matrices can easily be compared with each other. We call the matrix-based representation of BCMs *allocation models*. The term allocation refers to the transformation of BCMs into matrices. *Alignment models* represent the actual comparison of two allocation matrices. Figure 4 depicts the relationships among the three viewpoints, the new modelling abstractions of alignment and the existing abstractions for modelling BPs and IT.

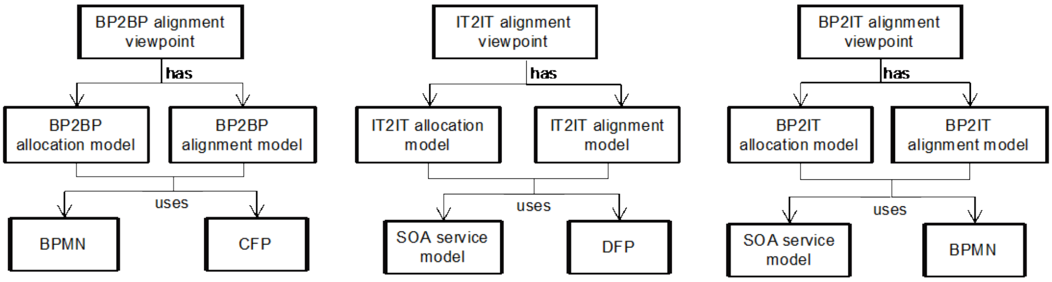

Figure 4: The model abstractions of the BITA alignment viewpoints*

In collaboration, organisations aim to realize a reference design they all can comply with. Compliance requires comparing two versions of a model. We therefore introduce the concepts *concrete* and *reference* models. Concrete models represent the current models (AS-IS or any new proposed designs) of the collaborating organisations; reference models represent the generic TO-BE models the organisations aim to comply with. The reference models describe the *reference architecture* and the concrete models describe the architectures adopted by the individual organisations (referred to as *concrete architectures*). Ideally alignment is modelled by comparing concrete allocation matrices with the corresponding reference allocation matrix. However, in practice, the required concrete allocation matrices are often unavailable (refer, for instance, to Stirna and Zdravkovic 2015). Therefore, concrete allocation matrices are determined from any available information sources, such as product descriptions. To specify the match or mismatch between allocation matrices we introduce the concept of *alignment attributes*. We use three distinct alignment attributes borrowed from the reflexion modelling approach (Murphy *et al.* 2001), which are: *convergence, divergence* and *absence.* The relationship between alignment attributes, reference models and concrete models is depicted in Figure 5.

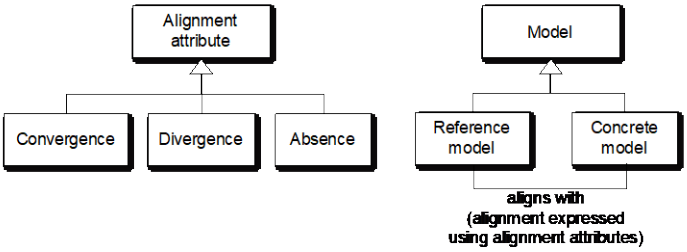

*Figure 5: The alignment attributes of BITA**

The three distinct alignment attributes are sufficient if organisations are compared pairwise. However, a reference matrix is compared with as many concrete matrices as there are organisations. This will lead to *partial convergence*, *partial absence* or *partial divergence*. However *partial absence* is the same as *partial convergence,* therefore*,* there are in total five alignment attributes. The alignment attributes are defined explicitly as follows:

Convergence: Consider a cell in a reference allocation matrix (for instance, that cell corresponds to an activity given in a row and an organisation given in a column), and that cell is assigned *true* (allocated). The reference allocation thus states that the specific organisation should execute the particular activity in a BCP model. If the corresponding cell in every concrete allocation matrix is also assigned *true*, then we assign the corresponding cell of the alignment matrix as *convergent.*

Absence: Consider a cell in a reference allocation matrix that is assigned *true* (allocated). If the corresponding cell in every concrete allocation matrix is assigned *false*, then we assign the corresponding cell of the alignment matrix as *absent*.

Divergence: Consider all corresponding cells in all corresponding concrete allocation matrices that are assigned *true* (allocated). If the corresponding cell in the corresponding reference allocation matrix is missing or assigned *false*, then we assign the corresponding cell of the alignment matrix as *divergent*.

Partial convergence (partial absence): Consider a cell in a reference allocation matrix that is assigned *true* (allocated). If the corresponding cells of some (but not all) of concrete allocation matrices are also assigned *true*, then we assign the corresponding cell of the alignment matrix as *partially convergent*.

Partial divergence: Consider a cell in any of the concrete allocation matrices that is assigned *true* (allocated). If the corresponding cell in the corresponding reference allocation matrix is missing or assigned *false*, then we assign the corresponding cell of the alignment matrix as *partial divergent* if it is not already assigned *divergent*.

For completeness, we also include an alignment attribute called *invalid* to indicate that the allocation is invalid or impossible in both the reference and the concrete matrices. The alignment attributes are summarized in Table 3.

Table 3: Alignment attributes.

Alignments	*Allocations*		*Notations*
	reference	*Concrete*	
Convergence	√	√	+
Absence	√	*x*	–
Divergence	*x*	√	~
Partial convergence	√	√ | x	±
Partial divergence	x	√ | x	#
Invalid	*x*	*x*	*x (or left empty)*

5.2 SYSTEMATIC APPROACH

Alignment of information systems across multiple organisations takes substantial effort and thus BITA* is meant to be applied interactively over a long period of time. In each iteration, the organisation specific concrete models proposed by the stakeholders' business analysts and software architects are compared with the corresponding reference models. Each iteration brings the stakeholders closer to consensus, which means they will propose concrete architectures that are closer to the new reference architecture—each iteration reduces the alignment gap.

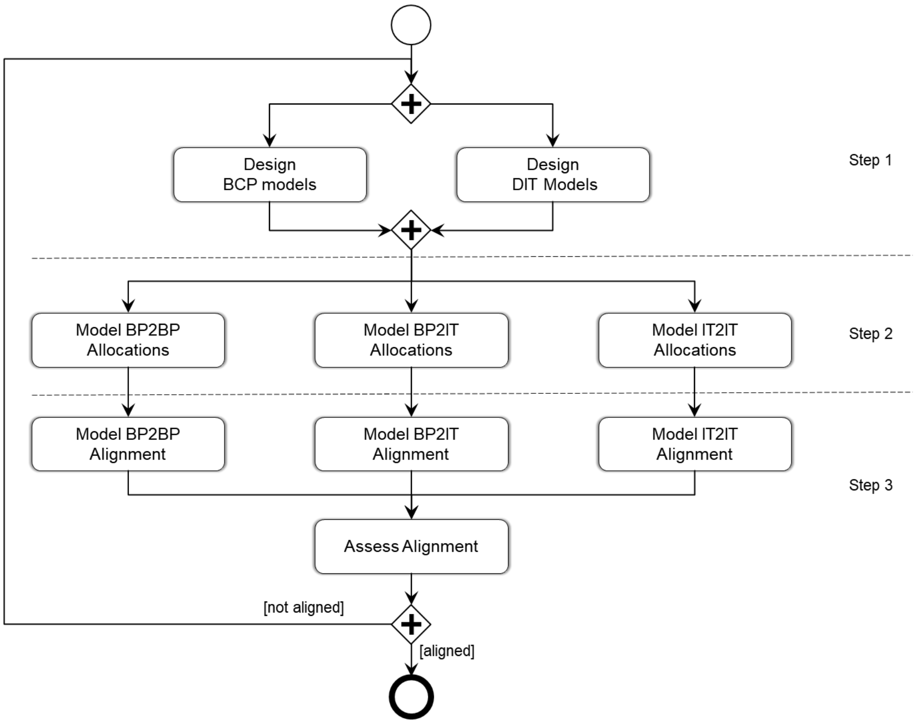

Figure 6: The BITA alignment process.*

The overall approach used in BITA* is depicted in Figure 6. The approach consists of three basic steps:

1. Design the Required Models

In this step the reference and concrete BCMs are designed. The BCMs are modelled using BPMN and SOA service models.

2. Model the Allocations

In this step the BP2BP, BP2IT and IT2IT reference allocation matrices are derived. Reference allocation matrices are derived from reference models. In parallel processes concrete allocation matrices are derived from concrete models. For each reference matrix there may be as many concrete matrices as there are organisations. It suffices to state here that the cells of allocation matrices are assigned a binary (*true*/*false*) value. In this paper we use a *tick mark* in a cell to indicate *true* (allocated), and a cell is *left blank* to represent *false* (not allocated).

3. Model the Alignments

In this step a reference allocation matrix is compared with the corresponding concrete allocation matrices. The alignment matrices are a copy of the corresponding reference allocation matrices extended with the new allocations that are only found in the corresponding concrete allocation matrices. The results of the comparison are indicated by alignment attributes, which are *convergence*, *absence*, *divergence, partial convergence* and *partial divergence*.

The steps below describe how a single alignment (a single cell in an alignment matrix) can be filled in with alignment attribute based on a round table discussion among a facilitator and stakeholders:

1. Present (or describe) the reference allocations to the stakeholders (representatives of the collaborating organisations) and ask for their views about the allocations.
2. If all of them agree with the reference allocation, then fill in *convergence* (+) in the corresponding cell of the alignment matrix.
3. If some of them agree, while others not, with the reference allocation, then fill in *partial convergence* (±) in the corresponding cell of the alignment matrix.
4. If all of them disagree with the reference allocation, then fill in *absence* (−) in the corresponding cell of the alignment matrix.
5. If all of them come up with an alternative allocation, include the cell in the alignment matrix, then fill in *divergence* (~) in the new cell of the alignment matrix.
6. If some (not all) of them come with an alternative allocation, include the cell in the alignment matrix, then fill in *partial divergence* (#) in the new cell of the alignment matrix.

5.3 ALIGNMENT VIEWPOINTS

In this section, we present the three viewpoints of BITA* and the corresponding modelling abstractions. In section 5.3.1 we present the BP2BP alignment viewpoint, in section 5.3.2 we present the BP2IT viewpoint, and in section 5.3.3 we present the IT2IT viewpoint.

5.3.1 BP2BP Alignment Viewpoint

The BP2BP alignment viewpoint provides a BP2BP allocation and alignment models. The BP2BP allocation model is used for representing BCPs in a matrix form.

5.3.1.1 BP2BP Allocation Model

Activities, control flows and organisations are key elements of BCP models. A BCP model, such as the one used in the case study (see Figure 14 in section 6), is essentially a specification of which organisation is responsible for which activity and how the control ‘flows’ from one activity to the next. We use CFPs in order to capture the control flows within BPMN models in a matrix form.

BPMN models can be represented as BP2BP allocation using two types of allocations: *activity allocations* and *CFP allocations*. An *activity allocation matrix* is a table that maps activities to organisations. Activity allocations are derived directly from a BPMN model by identifying the pool (and thus the organisation) and the activity that belongs to the pool. To derive a *CFP allocation*, both the activities and other CFPs that are contained in it must be identified and allocated to the CFP. The CFP allocation processes is described later in the case study with the help of an example. A *CFP allocation matrix* is a collection of CFP allocations. Figure 7 depicts the BP2BP allocation model.

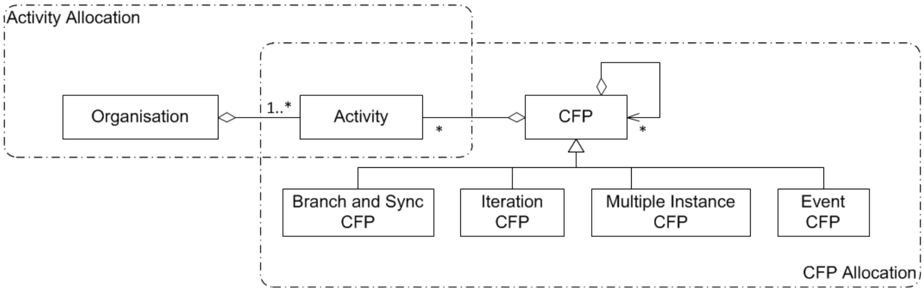

Figure 7. BP2BP allocation model

5.3.1.2 BP2BP Alignment Model

The BP2BP alignment model (see Figure 8) consists of *activity* and *CFP alignments* matrices, corresponding to the activity and CFP allocation matrices, respectively.

An activity alignment matrix is the result of comparing a reference activity allocation matrix with the corresponding concrete activity allocation matrices. Likewise, a *CFP alignment matrix* is the result of comparing a reference CFP allocation matrix with the corresponding concrete CFP allocation matrices. Each cell of an alignment matrix is assigned one of the five *alignment attributes*.

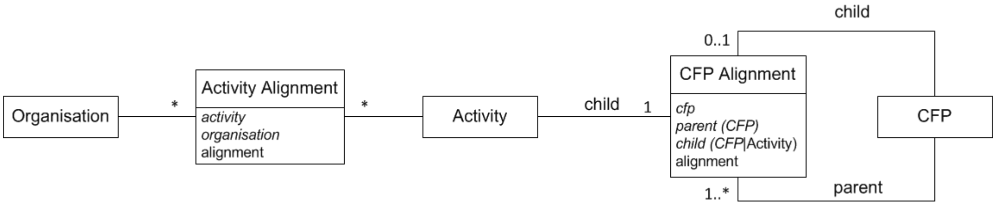

Figure 8: The BP2BP alignment model

5.3.2 BP2IT Alignment Viewpoint

The BP2IT alignment viewpoint provides a BP2IT allocation and alignment models.

5.3.2.1 BP2IT Allocation Model

Organisations and parts of BPs and IT systems (*i.e.* BPs, activities, data objects and IT services) are the key elements of BP2IT allocations. The BP2IT allocation model consists of two types of allocations: *IT service allocation* and *I/O (data input/data output) allocation. IT service allocations* describe the relationships among *IT services*, *clients (organisations who execute activities* using the IT services) and *providers* (organisations that support the IT services). An *IT service allocation matrix* is a collection of IT service allocations represented in a matrix form. *I/O allocations* describe the data inputs to and the data outputs from an activity. Obviously, an I/O allocation is either a *data input allocation* or a *data output allocation*. An *I/O allocation matrix* is a collection of all I/O allocations represented in a matrix form. Figure 9 depicts the BP2IT allocations model.

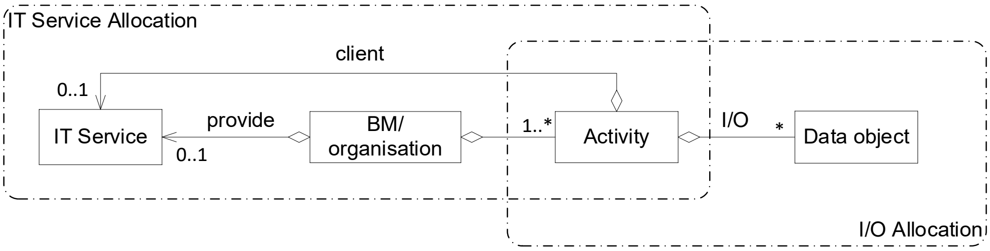

Figure 9: BP2IT allocation model

5.3.2.2 *BP2IT Alignment Model*

The BP2IT alignment model consists of *IT service* and *I/O alignment* matrices (see Figure 10), corresponding to the IT service and I/O allocation matrices, respectively. An *IT service alignment matrix* is the result of comparing a reference IT service allocation matrix with the corresponding concrete IT service allocation matrices. An *I/O alignment matrix* is the result of comparing a reference I/O allocation matrix with the corresponding concrete I/O allocation matrices. The I/O alignment matrix can be divided into two separate input data and output data alignment matrices.

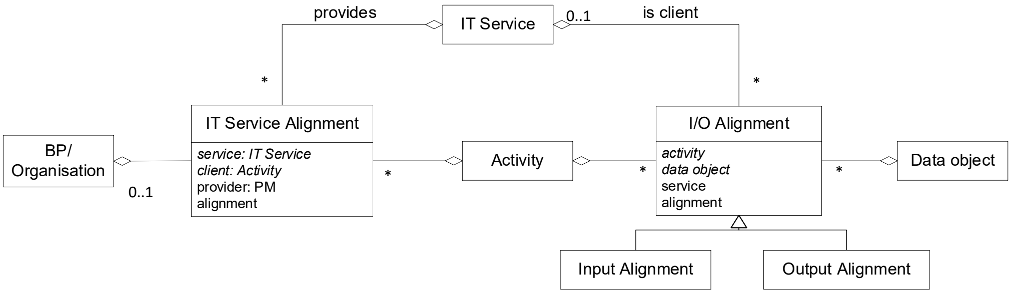

Figure 10: BP2IT alignment model

5.3.3 *IT2IT Alignment Viewpoint*

The IT2IT alignment viewpoint provides IT2IT allocation and IT2IT alignment models. The allocation model uses DFPs to represent data sharing as allocation matrices.

5.3.3.1 *IT2IT Allocation Model*

The IT2IT allocation model represents a DIT system. The aspects of DIT models considered in this paper are the distribution of IT systems and data objects, and how data are shared. The IT2IT allocation model consists of three types of allocations: *IT system allocations*, *data object allocations* and *DFP allocations*. *IT system allocations* describe which organisations in the collaboration network provides which IT systems, and how the IT services are distributed among the IT systems. An *IT system allocation matrix* is a collection of IT system allocations represented in a matrix form. *Data object allocations* describe which organisations in the collaboration network provide which data objects. A *data object allocation matrix* is a collection of data object allocations represented in a matrix form. *DFP allocations* describe how data objects are shared and used. A *DFP allocation matrix* is a collection of DFP allocations represented in a matrix form. A data object can be assigned up to four DFPs corresponding to the four data access and usage concerns, resulting in four DFP allocation matrices. Figure 11 depicts the IT2IT allocations model.

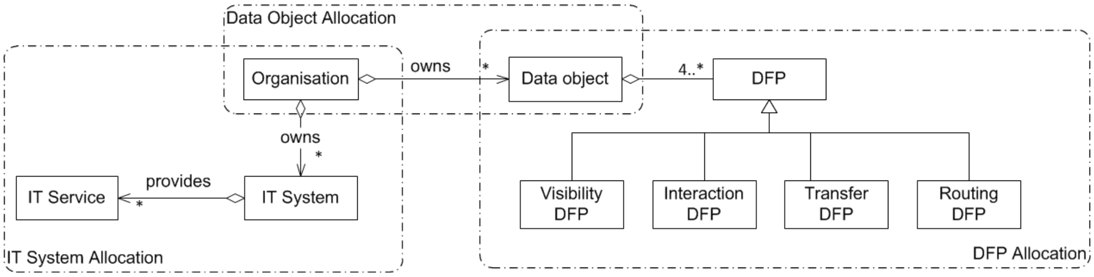

Figure 11: IT2IT allocation model

5.3.3.2 *IT2IT Alignment Model*

The *IT2IT* alignment model consists of *IT system, data object* and *DFP alignment* matrices (see Figure 12), corresponding to the IT system, data object and DFP allocation matrices, respectively. As before, each *alignment matrix* is the result of comparing the reference allocation matrix with the corresponding concrete allocation matrices.

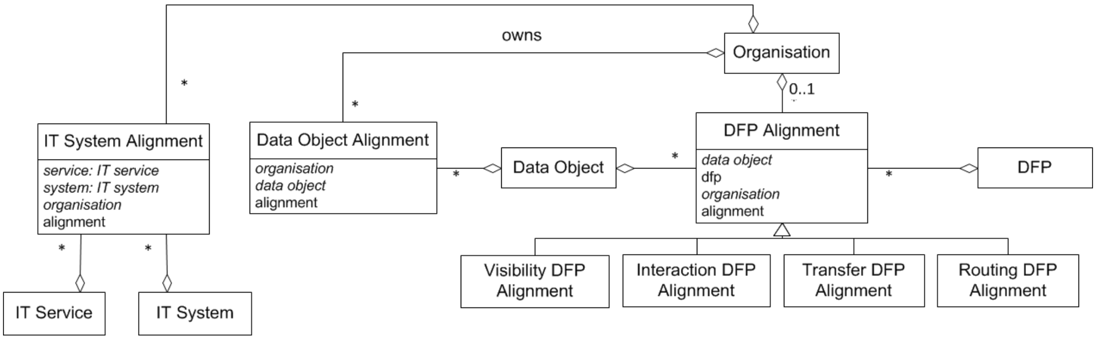

Figure 12: IT2IT alignment model

6 CASE STUDY VALIDATION

We now apply the BITA* approach to a meat supply chain case study. The BCP and DIT models of a number of collaborating organisations were aligned in order to create a new reference architecture for supply chain wide transparency system. The stakeholders involved include farmers, slaughterhouse and meat processing companies, retailers, standardisation bodies, and a third-party solution provider as described in our previous study (Kassahun *et al.* 2014). The case study and an example of a possible reference BCP model is presented in section 6.1. The corresponding reference architecture is described in section 6.2. An example of a concrete architecture provided by one of the collaborating organisations is presented in section 6.3. Finally, the application of BITA* is presented section 6.4.

6.1 TRANSPARENCY SYSTEM FOR MEAT SUPPLY CHAINS

A supply chain is a set of three or more entities that process and move products, services, finances and information between the primary suppliers and final customers (Mentzer *et al.* 2001). A meat supply chain consists of a network of food operators that transform slaughter animals into finished meat products. The primary suppliers include breeders and feed companies, while the final customers are largely end-consumers. An important concern in meat supply chains is how to provide chain-wide transparency in order to meet the requirements of safety, quality and consumer trust in meat products (Kassahun *et al.* 2014). To meet these requirements a transparency system must support the collaboration of the supply chain actors in order to share transparency information. The actors in meat supply chains can be categorized as food operators and third parties. Food operators include the farmers, meat processors, distributors and retailers. Third parties include regulators, inspectors, and laboratories. The collaboration involves the sharing of transparency data among the supply chain actors. A conceptual model for meat supply chain transparency systems is shown in Figure 13.

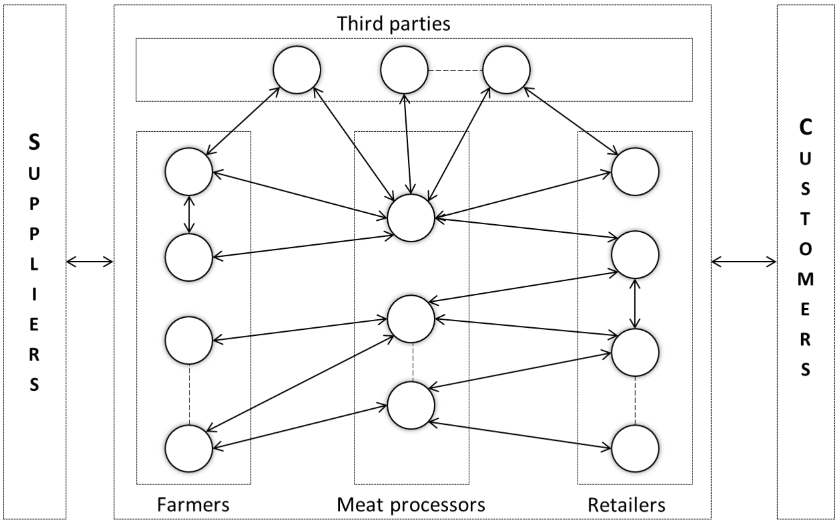

Figure 13: A conceptual model of meat supply chains.

To achieve chain-wide transparency the relevant BCPs and DIT of the collaborating actors (food operators and third parties) have to be aligned. To achieve alignment, at least two conditions have to be met: (1) there is a reference architecture that defines common BP and IT models in sufficient details, and (2) all actors comply with the reference architecture. A recognized global standard that aims to achieve the first goal is the Electronic Product Code Information System (EPCIS, EPCglobal 2014). EPCIS is a specification based on SOA developed by GS1. GS1 is a global consortium that designs global standards for supply chain transparency including the numbering system for barcodes that are used in virtually every consumer product and logistic package. Achieving the second goal, which is complying with the EPCIS standard, turned out to be infeasible for many supply chain actors, and therefore, an adapted version the reference architecture is required.

6.2 A GENERIC REFERENCE ARCHITECTURE

EPCIS represents a generic reference architecture that applies to all supply chains. EPCIS specifies a distributed network of enterprise transparency systems that are loosely connected through a discovery service. It provides reference BCP and DIT models.

6.2.1 A Reference BCP model

EPCIS provides generic reference *data capture* and *data query* BP models, which are the two key BP models in any transparency system. The data capture BP defines how transparency data should be scanned (or read by any other means) from each product item and stored in a transparency data repository. Here, the data primarily correspond to the events that are related to the physical movement or processing of products (such as loading, cutting and mixing). A data query BP defines how transparency data should be retrieved from transparency data repositories. According to the EPCIS specification, data capturing is a local process that is carried out independently by each food operator; and *data querying* is a BCP that links local BPs implemented by multiple food operators and third parties. In order to distinguish between local BPs, on one hand, and the integration of those local BPs into a BCP, on the other hand, distinction is made in the literature between *internal transparency systems* (*ITS*) that provide the ability to capture and query transparency data within an organisation, and *external transparency systems* (*ETS*) that provide the ability to query transparency data across the supply chain (Moe 1998, Gandino *et al.* 2009).

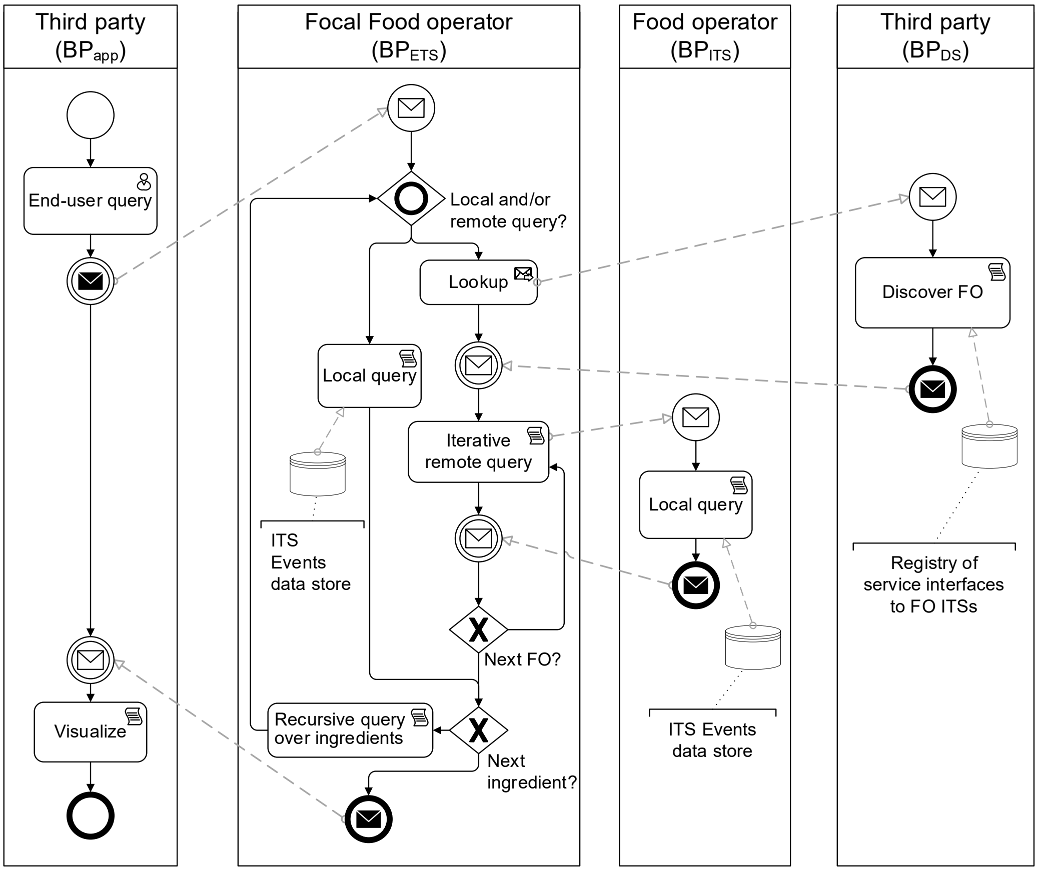

Figure 14: A query BCP according to the EPCIS reference architecture.

Different query BCPs can be defined based on the EPCIS reference architecture (Kürschner *et al.* 2008, Lorenz *et al.* 2011, Kywe *et al.* 2012). Figure 14 shows one possible query BCP. The model consists of four BPs that are executed by the respective actors. BP_{ITS} and BP_{ETS} represent the data query BP models that take place at the food operators of the supply chain. BP_{app} represents the BP implemented in an end-user software application (*app*), which receives the request for transparency data from the end user and presents the query result to the user in an intuitive and user-friendly manner. The reference architecture does not specify who should provide such an app but in our specific case the provider is is an external third party. The app triggers BP_{ETS} of a food operator. BP_{ETS} is provided by what we here refer to as the *focal* food operator—*focal* because it receives the request for transparency data on behalf of the supply chain. Also note that the term *focal* is not a permanent role but is rather valid only for the given request. In this specific case the focal food operator realizes external transparency by retrieving transparency data across the supply chain. It does so by recursively querying for transparency data locally and externally (from the transparency systems of all other food operators). The subscript ETS indicates, therefore, that BP_{ETS} realizes external (chain-wide) transparency. BP_{ITS} retrieves transparency data locally, from the local EPCIS repository and thus realizes internal transparency.

The focal food operator discovers the addresses of its partner food operators from a registry maintained by a third-party, which is not necessarily the same third party that provides the app. The discovery BP, implemented in a Discovery Service (DS), is represented by the model BP_{DS}. Given an ID of a product item, the discovery service provides a list of *URLs* representing ITSs from where transparency data can be queried. In practice, BP_{ETS} and BP_{DS} can be complex models for which diverse approaches are proposed (Kürschner *et al.* 2008, Lorenz *et al.* 2011, Kywe *et al.* 2012).

Finally, end-users use an app to scan the ID of a product item and retrieve associated transparency data about the product item. For simplicity, we assume that each meat product item has a unique ID printed as barcode which end-users can scan. BP_{app} combines the outputs from BP_{ETS} with product descriptions (retrieved from a master data repository) into understandable and user-friendly information and presents it to the user (for detailed description refer to Kassahun *et al.* (2014) and Kassahun *et al.* (2016)).

It is important to note that in this query BCP model the details of the internal BPs are not provided. Only those activities that are candidate for alignment—because they can potentially be assigned to a different organisation—are included. Such activities are the concern of the alignment process.

6.2.2 A reference IT model

The EPCIS reference architecture specifies two basic IT services: data capture IT service (CaptureSrv) and data query IT service (QuerySrv). The CaptureSrv service corresponds to the data capture BP. Data capture is a local BP (and not a BCP); therefore, CaptureSrv is not available to collaboration partners. There are two types of query services, $\text{QuerySrv}_{\text{ITS}}$ and $\text{QuerySrv}_{\text{ETS}}$, that correspond to BP_{ITS} and PM_{ETS}, respectively. The reference architecture does not describe how the $\text{QuerySrv}_{\text{ETS}}$ should be composed from the distributed $\text{QuerySrv}_{\text{ITS}}$ services. The $\text{QuerySrv}_{\text{DS}}$ service corresponds to the BP_{DS}. $\text{QuerySrv}_{\text{app}}$ realizes the BP_{app}.

The IT services fulfil one or two of the SOA roles. $\text{QuerySrv}_{\text{ETS}}$ is a *client* of $\text{QuerySrv}_{\text{DS}}$ and $\text{QuerySrv}_{\text{ITS}}$ services. In turn, $\text{QuerySrv}_{\text{app}}$ is a *client* of $\text{QuerySrv}_{\text{ETS}}$. In both cases, the latter are said to be *provider* of IT services to the former. At any one time, a food operator provides either $\text{QuerySrv}_{\text{ETS}}$ or $\text{QuerySrv}_{\text{ITS}}$. Third parties that provide $\text{QuerySrv}_{\text{DS}}$ are *service brokers*.

The reference architecture defines also a data model for transparency systems, which is referred to as the EPCIS event model. An *event* data object is an aggregate data object that contains four data objects called event dimensions. They are conveniently called the *what*, the *when*, the *where* and the *why* of events. The *what* data object is an ID and represents the unique identification of a product item the event is about. The *when* data object is a timestamp and represents the date and time the event occurred. The *where* data object is an ID of the place where the event occurred. And, the *why* data object is a predefined vocabulary of *reasons* for recording the event and the resulting *state* of the product item. IDs and predefined vocabularies are largely meaningless to human readers. The descriptive information corresponding to IDs and vocabularies are retrieved from master data repositories (GS1 2014) by the app. Yet another data object is a service URL (*srvURL*) that identifies the web address of a $\text{QuerySrv}_{\text{ITS}}$.

6.3 A CONCRETE ARCHITECTURE

Figure 15 provides an example of an informal description of a concrete architecture provided by one of the meat processors of the case study. It represents just one of the many concrete architectures of the different collaborating organisations. The architecture is not only different from the architectures adopted by other supply chain actors but also different from the reference architecture. For instance, according to the reference architecture data capturing is a local BP and data querying is a collaborative process (a BCP). In the concrete architecture, however, the meat operator offers its data capture IT services to its collaboration partners (mainly farmers), and thereby makes it a BCP. Besides, a number of new IT services involving third parties, such as QS (a quality assurance agency) and HIT (a national bovine animal registration office) are introduced.

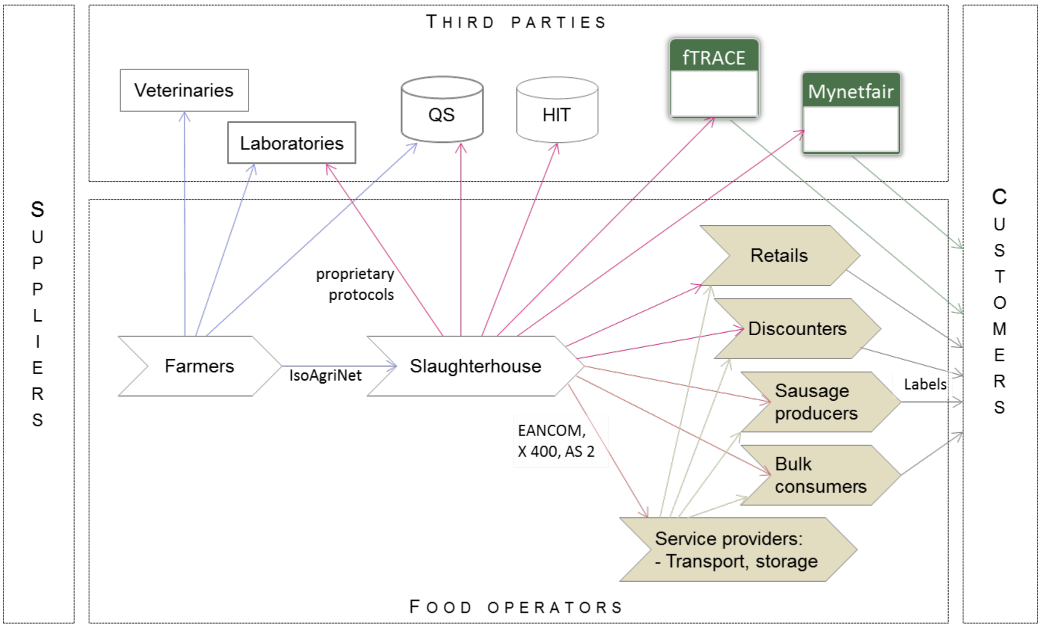

Figure 15: Concrete architecture of a chain-wide transparency system according to a large slaughterhouse in Germany (Kassahun et al. 2014). QS (Qualität und Sicherheit) represents a database maintained by German meat industry initiated quality assurance organisation (QS 2013). HIT (Herkunftssicherungs- und Informationssystem Tiere) represents the German national database for registration of movement of bovine animals (HI-Tier 2013). Mynetfair represents a trade fair web portal and the associated mobile app (Mynetfair 2013). fTRACE represents a meat transparency system and the associated mobile app (fTRACE 2013).

6.4 MODELLING ALIGNMENTS

The desired chain-wide transparency would be realized if each supply chain actor complies with the BCMs of the reference architecture, which is the generic EPCIS reference architecture. Unfortunately, many of the supply chain actors did not, and cannot, comply with the generic reference architecture; *i.e.* they cannot adopt and BCMs derived from the reference architecture as they are. There are three basic reasons for this. First, several European food regulations impose conflicting requirements on transparency systems. For instance, regulations on the movement and slaughter of bovine animals (EC 2000, EC 2004) mandate a different BCP model than the BCP model the General Food Law regulation suggests to traceability of meat products (EC 2002). Therefore, different BCMs apply to farmers and to meat processors. Second, some large food operators have already expensive legacy transparency systems in place that apply again different BCMs. Third, many other supply chain actors do not have the resources to deploy the required IT systems (Kassahun *et al.* 2014). As a result, the BCP and DIT models in place in meat supply chains do not comply with the reference models given in section 6.2. However, new market circumstances (Kassahun *et al.* 2014) had made it necessary to move away from the concrete architecture such as that depicted in section 6.3. The meat sector in general, and the collaborating organisations of the case study in particular, need to derive a new reference architecture that they can comply with.

The new reference architecture should build on an existing generic reference architecture, and the EPCIS standard was chosen as appropriate generic reference architecture. However, the models described in sections 6.3 (concrete BCMs) and in section 6.2 (reference BCMs) are not easy to compare with each other, let alone align them. In the following section we show how the BITA* alignment framework is used to compare and align BCMs using the reference BCMs described in sections 6.2 and the concrete BCMs described in sections 6.3 as a starting point. A detailed description of the alignment of some of the concrete BCMs and the EPCIS-based reference BCMs that contributed to the development of the alignment framework is provided in our previous research (Kassahun *et al.* 2014, Kassahun *et al.* 2016).

6.4.1 BP2BP View

In this section we present the BP2BP allocation matrices derived from the reference BCP model depicted in Figure 14. We then present the BP2BP alignment matrices that are derived from the reference BP2BP allocation matrices and the available information that describes the concrete BCP models.

6.4.1.1 BP2BP Allocations

Table 4 shows the reference activity allocations following the BCP model depicted in Figure 14. Deriving activity allocations is straightforward—activities are listed as rows and organisations (pools) are listed as columns. The columns and rows can directly be read from the BCP model depicted in Figure 14.

Table 4: Reference activity allocations

Activities	*Organisations*	
	Food operators	*Third party*
a1: end-user query		√
a2: decide where to query	√	
a3: local query	√	
a4: lookup		√
a5: iterative remote query	√	
a6: recursive query over ingredients	√	
a7: visualize		√

To model the CFP allocations, the CFPs have to be identified from the BCP model given in Figure 14. It is easier to identify CFPs from a BPMN choreography model than a BPMN collaboration model. Therefore, we derive in Figure 16 the choreography model version of the collaboration model given by Figure 14. The CFPs are depicted as overlapping blocks (*p1* to *p8*).

Table 5 depicts the resulting reference CFP allocation matrix. A CFP allocation matrix is a three-dimensional matrix represented as a two-dimensional table. (We will also hereafter roll up all multidimensional matrices into two-dimensional tables.). Pattern *p1* is a *sequence* (*cfp-1*) CFP since the three patterns (*p2*, *p3, p4*) and the two activities (*a1*, *a7*) contained in it are arranged sequentially. Patterns *p2* is a *transient trigger* (*cfp-23*) event CFP; *p3* is an *implicit termination* (*cfp-11*) event CFP. Pattern *p4* is a *recursion* (*cfp-20*) CFP since activity *a6* recursively triggers its containing CFP pattern *p4*. Pattern *p5* is a multi-choice (*cfp-6*) CFP since one or both of the two parallel paths can be executed. The parallel paths merge in pattern *p6*, which is multi-merge (*cfp-8*) CFP. Pattern *p7* is a sequence (*cfp-1*) CFP since the lookup activity (*a4*) and pattern *p8* are arranged sequentially. Pattern *p8* is a s*tructured loop* (*cfp-21*) CFP since remote queries are initiated iteratively.

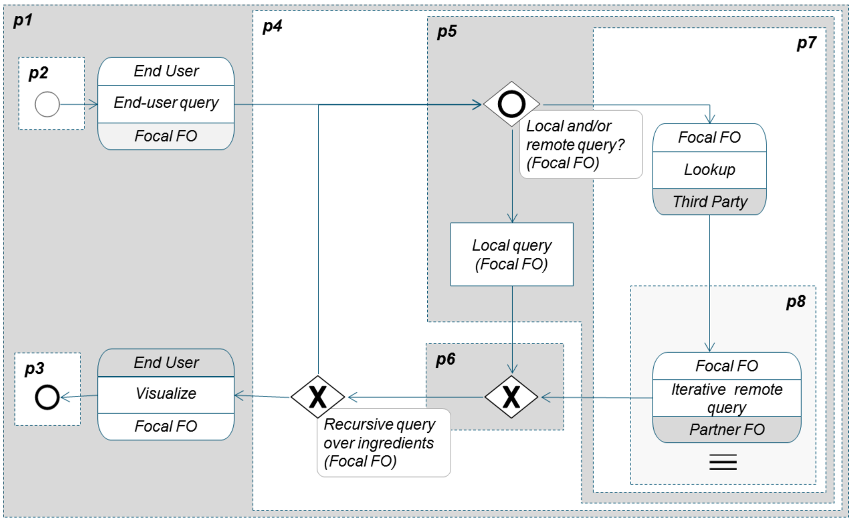

Figure 16: A CFP diagram for the query BCP model

Table 5: Reference CFP allocations

CFP			*parent*								*children*						
p_{id}	*id*	*name*	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*a1*	*a2*	*a3*	*a4*	*a5*	*a6*	*a7*
p1	*1*	*sequence*									√						√
p2	*23*	*transient trigger*	√														
p3	*11*	*implicit termination*	√														
p4	*22*	*recursion*	√													√	
p5	*6*	*multi-choice*				√						√	√				
p6	*8*	*multi-merge*				√											
p7	*1*	*sequence*					√							√			
p8	*21*	*structured loop*							√						√		

6.4.1.2 BP2BP Alignments

In order to model the BP2BP alignments, each collaborating organisation should, ideally, produce the concrete allocation matrices corresponding to the reference allocation matrices given in Table 4 and in Table 5. In practice, and also in this case, most organisations do not have the formal models of their BCPs and DIT, and as a result not all required concrete allocation matrices are available. The alignment matrices given in Table 6 and Table 7 are, therefore, derived following the systematic approach described in section 5.2 for addressing such shortcomings by comparing the two reference allocations with the best available concrete allocations, based om the models described in section 6.3.

The activity alignment matrix shown in Table 6 contains a new row and three new columns because the analysis of the analysis of the informal concrete architectures presented by the stakeholders showed that there is an additional activity and three new organisation types are involved in the concrete BCPs. Thus, in Table 6, the last row (*a8*: *data capture*) and three distinct types of food operators (*Farmers*, *Meat processes* and *Bulk customers*) were added. The three distinct food operator types in Table 6 replace the single food operator type used in the reference model (Table 4) because the three food operators follow three different BCPs.

We demonstrate how the alignment attributes are assigned by using the activities *a1 (end-use query)*, *a2 (decide where to query)* and *a8 (data capture)* which are associated with the *convergence*, *absence* and *divergence* alignment attributes, respectively. The allocation of activity *a1* to a third party is in *convergence* (+). The activity was allocated to a third party both in the reference and in the corresponding concrete BCP models. In fact, for the given case study this allocation is indeed supported in the form of the *fTrace* system that is provided by external third party, also called fTrace. The allocation of the decision activity *a2* to food operators is *absent* (–). This activity is allocated to food operators in the reference model, but is not supported in the concrete models. The activity *a8* (data capture across food operators) is a new activity that is not present in the reference model. Therefore, *a8* represents divergent (~) behaviour. However, since not all meat processors support *a8*, the allocation of *a8* to meat processors is only *partially divergent* (#). The rest of the cells of Table 6 are filled in a similar fashion.

Table 6: Activity alignments (there is only a single third party)*

Activities	*Organisations*			
	Food operators			
	Farmers	*Meat processors*	*Bulk customers*	*Third party (*)*
query process				
a1: end-user query				+
a2: decide where to query	–	–	–	*x*
a3: local query	+	+	+	~
a4: lookup	*x*	*x*	*x*	–
a5: iterative remote query	–	–	–	*x*
a6: recursive query (ingredients)	–	–	–	~
a7: visualize	*x*	*x*	*x*	+
data capture process				
a8: data capture	*x*	#	*x*	~

Table 7 shows the how reference CFPs are aligned with their concrete counterparts. No new patterns were identified in concrete query BCP models; therefore, no divergent CFPs are included. Pattern *p1, p2* and *p3* converge (+), the pattern *p4* largely converges except that it includes an additional allocation of *a3* to the pattern in the concrete allocation model. The rest of the CFPs are missing in the concrete models since the corresponding activities are absent.

Table 7: CFP alignments

CFP		*parent alignment*								*children alignment*						
p_{id}	*id*	*p1*	*p2*	*p3*	*p4*	*p5*	*p6*	*p7*	*p8*	*a1*	*a2*	*a3*	*a4*	*a5*	*a6*	*a7*
p1	*1*									+						+
p2	*23*	+														
p3	*11*	+														
p4	*22*	+										~			+	
p5	*6*				–						–	–				
p6	*8*				–											
p7	*1*					–							–			
p8	*21*							–						–		

6.4.2 BP2IT View

In this section we present the BP2IT allocation and alignment matrices derived from the reference and concrete BCP-DIT models described in sections 6.2 and 6.3.

6.4.2.1 BP2IT Allocations

Table 8 presents the reference IT service allocation matrix showing the relationships among *IT services, clients* (that execute *activities* with the support of IT service) and *providers* (that support IT services, including IT services that automate entire *BPs*). The allocations are derived from Figure 14 and the EPCIS IT services discussed in section 6.3.

Table 8: Reference IT service allocations (FO: food operator, 3P: third party)

IT service	*BP*				*SOA role*						
					Provider		*Broker*		*Clients*		
	App	*ETS*	*ITS*	*DS*	*3P*	*FO*	*3p*	*FO*	*3p*	*FO*	*End user*
$QuerySrv_{app}$	√				√						√
$QuerySrv_{ETS}$		√				√			√		
$QuerySrv_{ITS}$			√			√				√	
$QuerySrv_{discovery}$				√	√			√		√	

The reference I/O allocation matrix given in Table 9 shows how input and outputs data objects are allocated to activities. Here, only three data objects are considered out of a large number of data objects. In most activities an ID data object is needed since transparency requires identification of product items. Likewise, most activities return a list of event data objects (*i.e.* transparency information is returned). The *lookup* (*a4*) activity takes an ID and returns a list of *srvURLs*. The *recursive query* (*a6*) activity takes an event data object and returns the list of the IDs of the ingredients of the product—if there are any. Activities involved in remote queries (*a1* and *a5*) require *srvURL* as input, in addition to IDs. Visualization (*a7*) requires event data objects as inputs and master data (not included), and produces information to end-users, which is not modelled as a data object.

Table 9: Reference I/O allocations

Activities	*Inputs*			*Outputs*		
	ID	*event*	*srvURL*	*ID*	*event*	*srvURL*
a1: end-user query	√		√		√	
a2: decide where to query	√				√	
a3: local query	√				√	
a4: lookup	√					√
a5: iterative remote query	√		√		√	
a6: recursive query (ingredients)		√		√		
a7: visualize		√				

6.4.2.2 BP2IT Alignments

The BP2IT alignments given in Table 10 and Table 11 are derived in the same manner as BP2BP alignments.

Table 10: IT service alignments (RG: (national) food regulatory agency, V/L: (national) veterinary/laboratory registration system, MP: online marketplace)

IT service	*BP*					*SOA role*									
						Provider					*Broker*		*Clients*		
	App	*ETS*	*ITS*	*DS*	*BC*	*3P*	*FO*	*RG*	*V/L*	*MP*	*3p*	*FO*	*3p*	*FO*	*End user*
$QuerySrv_{app}$	+					+									+
$QuerySrv_{ETS}$		–				~	–						~		
$QuerySrv_{ITS}$			+				±							–	
$QuerySrv_{discovery}$				–		–						–		–	
$CaptureSrv_{FO}$					~		~								
$CaptureSrv_{fTrace}$					~	~									
$CaptureSrv_{QS}$					~			~							
$CaptureSrv_{HIT}$					~			~							
$CaptureSrv_{VET}$									~						
$CaptureSrv_{LAB}$									~						
$CaptureSrv_{Mynetfair}$					~					~					

We describe how alignment attributes are assigned using example *convergent*, *divergent* and *absent* IT service alignments. In the reference BCP-DIT model $QuerySrv_{app}$ IT service was provided by *3Ps* and used by end users (*clients*). For the given case study these allocations were indeed supported in the form of the *fTrace* app *provided* by a *3P* and used by *clients*. Both allocations are *convergent* (+). In the reference BCP-DIT model $QuerySrv_{ETS}$ IT service was provided by food operators (*FO*) and used by other *FOs* (*clients*) but these allocations were largely absent and, in some cases, divergent for the given case study. The allocation of $QuerySrv_{ETS}$ to BP_{ETS} is *absent* (–) because $QuerySrv_{ETS}$ implements a different process model than BP_{ETS}. The allocation of $QuerySrv_{ETS}$ to provider *FO* is *absent* (–) because the FOs are not providing $QuerySrv_{ETS}$; instead, a *3P* does, thus *divergent* (~). The allocation of $QuerySrv_{ETS}$ to client *3P* is *divergent* (~) because *client 3P* is not using *provider FOs* but *provider 3P* (potentially a different 3P than the client self). Note also that all capture IT services are new and thus *divergent* (~).

Table 11: I/O alignments.

Activities	*Inputs*			*Outputs*		
	ID	*event*	*srvURL*	*ID*	*event*	*srvURL*
a1: end-user query	+		+		+	
a2: decide where to query	–				–	
a3: local query	+				+	
a4: lookup	–					–
a5: iterative remote query	–		–		–	
a6: recursive query (ingredients)		+		+		
a7: visualize		+				
a8: data capture					~	

The I/O alignment matrix shown in Table 11 is derived as follows. In the reference BCP-DIT model, the activity *a1* takes *ID* and *srvUrl* data objects and yields events. This is consistent with how the concrete allocations are—and how the *fTrace* app works for the given case study. Therefore, all the three allocations with reference to *a1* are *convergent* (+). The allocations in relation to the activity *a2* are *absent* (–) because the activity itself is *absent* (–, see Table 6). The allocations in relation to the activity *a8* are *divergent* (~) because the activity itself is *divergent* (~, see Table 6).

6.4.3 IT2IT View

In this section we present the IT2IT allocation and alignment matrices derived from the reference and concrete DIT models described in sections 6.2 and 6.3.

6.4.3.1 IT2IT Allocations

IT2IT viewpoint specifies *IT system*, *data object* and *DFP allocations*. Since neither the reference architecture nor the descriptions of the concrete architectures provide information about IT system models, IT system allocations and the corresponding alignments are not included.

Table 12: Reference data object allocations.

Data objects	*Organisations*	
	Food operators	*Third party*
ID	√	√
event	√	
srvUrl		√

The reference data object allocations representing how the reference data objects are allocated to organisations are presented in Table 12. Though a great number of data objects, particularly involving master data, may be involved, we considered only products identifications (*IDs*), transparency data items (*events*) and IT service addresses (*srvURLs*), which are the three key data objects of the reference architecture. Their allocation is simple: food operators manage their own event data objects; the third party manages the IT service addresses. Both actors manage IDs for different purposes in the query PMs: food operators resolve ID to events, while the third party resolves ID to srvURLs.

Table 13: DFP allocations.

Categories of DFPs	*DFPs*	*Data objects*		
		ID	*event*	*srvURL*
Visibility	*1*	√		
	8		√	√
Interaction	*9*	√		
	16		√	√
Transfer	*27*	√	√	√
	28	√	√	√
Routing	*36*			√
	40		√	

Table 13 shows the reference DFP allocation matrix involving *data objects* and *DFPs.* According to the reference architecture the allocation of DFPs to data objects is not dependent on the organisation; therefore, the reference allocation matrix does not include the organisation dimension. The allocation matrix can potentially consist of a maximum of 40 rows, one for each DFP. A data object is associated with at least four DFPs; one from each category of DFPs. The reference DFP allocations are discussed based on the four categories.

Visibility: IDs are allocated *activity scope* (*dfp1*) DFP because an ID is obtained from end-user and is passed from one activity to the next. Events and srvURLs are, in comparison, allocated *external data scope* (*dfp-8*) DFP because they are fetched from repositories that are external to the process orchestration system.

Interaction: IDs are allocated *activity-to-activity* (*dfp*-9) DFP because IDs are passed from activity to activity. Events and srvURLs are allocated *activity pulls data* (*dfp-16*) DFP because activities pull data from external EPCIS repositories.

Transfer: All data objects are allocated *pass inputs by value* (*dfp-27*) and *pass outputs by value* (*dfp-28*) transfer DFPs.

Routing: IDs do not affect the routing of the control flow; therefore, no routing DFPs are assigned to them. *Events* are assigned *data-based routing* (*dfp-40*) DFP because the content of an event data object determines if recursive queries are executed. S*rvURLs* are allocated *value as activity post-condition* (*dfp-36*) routing DFP since without a service address external queries cannot be executed.

6.4.3.2 IT2IT Alignments

The data object alignment matrix is simple. The allocations of ID and event data objects to organisations are *convergent* (+) since for the given case study these data objects are allocated as defined in the reference data allocation matrix. The allocation of srvURL to third party organisation is absent (–) since for the given case study there is only a single srvURL value. The data object alignment matrix is trivial and, therefore, it is not presented.

Table 14: DFP alignments.

Data flow concerns	*DFPs*	*Data objects*		
		ID	*event*	*srvURL*
Data query business collaboration				
Visibility	*1*	+		
	8		+	–
Interaction	*9*	–		
	16		–	–
Transfer	*27*	–	–	–
	28	–	–	–
Routing	*36*			–
	40		–	
Data capture business collaboration				
Visibility	*1*			
	8		~	
Interaction	*9*			
	16		~	
Transfer	*27*		~	
	28		~	
Routing	*36*			
	40			

The DFP alignment matrix shown in Table 14 is derived from the reference DFP allocation matrix given in Table 13. We describe how alignment attributes are assigned using example *convergent*, *divergent* and *absent* DFP alignments. In the reference data sharing model, IDs are allocated *activity scope* (*dfp1*) DFP. For the given case study, the IDs have activity scope. Therefore, the given DFP allocation is *convergent* (+). In the reference data sharing model events are allocated *activity pulls data* (*dfp-16*) interaction DFP. For the given case study, the events are not pulled from external transparency data repository. Note that, in the given case study, the focal food operator captures all transparency data and serves query requests from own local repository. Therefore, the given DFP allocation is *absent* (–). In the reference data sharing model data capture is a local process. For the given case study data capture is a collaboration process. All DFP allocations associated with new collaboration processes are considered *divergent* (~).

7 DISCUSSION

In this paper we proposed alignment as an approach for deriving new reference architectures from existing generic reference architectures and focused on the alignment concerns of multiple collaborating organisations. To address the alignment concerns more explicitly we have offered a business-IT alignment framework called BITA* and demonstrated the framework in a real-life business case. We distinguish three types of alignment concerns: BP2BP, IT2IT and BP2IT. BP2BP refers to the alignment of BCP models. IT2IT refers to the alignment of the models for the distributed IT. BP2IT refers to the alignment of the models that define how the BCPs should be supported by DIT.

Our approach can be best described by considering BP2BP alignment concerns, which deal with designing inter-organizational BPs. The extensive literature available indicates that designing inter-organizational BPs requires matching elements of the BPs of the collaborating organisations (Dijkman *et al.* 2009, Zhao *et al.* 2009, Weidlich *et al.* 2011, Antunes *et al.* 2015). Dijkman *et al.* (2009) formulates the problem of BP model alignment succinctly as: "…given a pair of BP models, determine which elements in one model are related to which elements in the other model". This problem is sometimes simple; for instance, the activity "send request" of one organisation is obviously related to the activity "receive request" of another organisation. However, match making among BPs is generally very difficult and can be compared with solving a picture puzzle with many missing and incorrect tiles.

Alignment requires matching business collaboration models, BCMs. The existing literature is based on matching BCMs from few (usually two) organisations with each other. Our approach is based on matching BCMs from many organisations with reference BCMs. To do so, each organisation should ideally provide BCMs that have to be aligned. In terms of the picture puzzle metaphor, to align, we need to compare partial picture puzzle solutions that are brought by different organisations with a reference picture puzzle solution. When only few organisations are involved and they are able to provide completed partial solutions, the methods proposed by existing literature, such as Antunes *et al.*, suffice. We propose BITA* for situations where many organisations are involved and the provided partial solutions have many missing parts. In this respect, this paper complements the existing literature by addressing novel concerns that were not explicitly addressed before.

To support the alignment process we provided required alignment modelling abstractions. An important aspect of the alignment modelling is syntactic and semantic comparability of models. To address these two issues, we represented BCMs as matrices. We adopted workflow patterns in order to convert graphical representation of BCMs into matrices. Matrices are easier to compare with each other than graphical representations. These matrix-based modelling abstractions for alignment were organised in BP2BP, BP2IT and IT2IT viewpoints following the ISO/IEC/IEEE (ISO/IEC/IEEE 2011) formal viewpoint design guideline.

We have developed the framework while collaborating with partners in large EU sponsored Future Internet (FI-PPP 2013) research programme. We were involved in various capacities, including requirements gathering, reviewing of projects, designing collaborative systems, and developing reference architectures. We have observed that practitioners in the agri-food sector already applied most of the alignment models, though informally and implicitly, when driving reference architectures (Steinberger et al. 2009, Verdouw et al. 2010, Wolfert et al. 2010, Kruize et al. 2016). Most of the modelling abstractions we proposed are thus rather straightforward. The exceptions are CFP and DFP alignment models, which are entirely new. In this respect, BITA* provides formal and comprehensive set of alignment models. While developing the approach and applying it to the case study within the FI-PPP programme, we could observe the following. First, adopting explicit models of alignment is very helpful for the facilitators of the alignment because it makes the alignment problems explicit and likewise creates a common understanding among the stakeholders. It also makes the alignment process efficient, transparent and auditable. If no explicit design abstractions are used, the possibilities for communicating alignment concerns and eventually addressing them becomes seriously limited. Thanks to the explicit alignment modelling, problems can be more easily identified and the relevant reference architecture adapted before the collaborating partners start the difficult process of redesigning their BPs and IT systems. This is vital because misalignment identified later in the process of system development would be more problematic and costlier for all stakeholders. Second, we observed that the representatives of the collaborating organisations often fail to produce explicit models of BCMs during the alignment process. They find it easier to describe the models adopted within their organisations based on models provided by the facilitators. Therefore, reference BCMs are an important starting point for expressing concrete BCMs.

We have focussed in this paper on alignment concerns of multiple collaborating organisations. The BITA* approach is, however, equally applicable when only two or few organisations are involved, in which case the collaboration architecture adopted by one of the organisations serves as a reference architecture. This is usually the case when dependent organisations (for instance, small food operators that supply large retailers) must align their BPs and IT systems with the dominant (focal) organisation (such as a large manufacturer or a large retailer).

The framework is applied in a real-life business case study. Case studies are susceptible to various validity threats. The major threats to validity are construct validity, internal validity and external validity (Yin 2003). Various strategies for mitigating these threats, including prolonged involvement to enhance shared understanding and triangulation inputs from different informants (Runeson *et al.* 2012), have been applied in this study as described in the two associated papers (Kassahun *et al.* 2014, Kassahun *et al.* 2016). These mitigation strategies are used to address construct and internal validity. External validity entails that, in order to claim the generality of the results, the study should be conducted in different contexts. This ideally requires multi case study approach (Yin 2003). However, we were not able to address this validity threat adequately because organising additional large design-related case studies involving multiple organisations is very difficult. Therefore, though we propose the BITA* framework as a generic framework that is equally applicable to domains, the development of the framework is largely influenced by the concerns we were faced within the agri-food sector, and thus the modelling constructs may not cover some of alignment concerns that occur in other application domains. Nevertheless, a thorough evaluation of such a framework requires a multi case study. Therefore, the framework needs to be evaluated further in future research in multiple case studies coming from various application domains to strengthen its validity.

8 CONCLUSION

In this paper we have presented BITA*, a framework for aligning business collaboration processes and the underlying distributed IT system of multiple collaborating organisations. The framework supports the development of reference architecture that consists of reference models of business collaboration processes and the corresponding distributed IT. We identified the process of developing a reference architecture as a problem of alignment between the existing concrete architectures adopted by the collaborating organisations, on one hand, and suitable generic reference architectures, on the other. We classified the alignment problem into three types of alignment concerns of the three types of stakeholders that are involved in the development of reference architectures. The concerns are business process to business process (BP2BP), IT to IT (IT2IT) and business process to IT (BP2IT) alignment concerns. The concerns are addressed by applying modelling abstractions we provided within three alignment design viewpoints as part of a design framework we called BITA*. The stakeholders that the viewpoints target are business analysts, IT specialists, and interdisciplinary teams of business analysts and IT specialists. We also provided a systematic approach for using the viewpoints and demonstrated the approach in a case study. A design framework for developing reference architectures has been lacking and BITA* aims to address this deficit in the existing body modelling abstractions for information systems.

Recognizing the difficulty of comparing business collaboration process and IT models, which mainly are graphical, we introduced a number of key concepts in BITA*. First, we used generic reference business collaboration models as a common set of models with which diverse models from diverse organisations can be compared. Then we introduced allocation models as means of uniformly representing diverse business collaboration process and IT models that have to be aligned. Third, we used workflow patterns to support capturing complex business collaboration process and IT models as allocation models. Allocation models are tabular models thus they can easily be compared with each other. These conceptualizations enabled us to design alignment models. The alignment models include explicit alignment attributes—convergence, absence, divergence, partial convergence, and partial divergence that can be assigned to allocations.

We presented a step-by-step approach that shows how business analysts and software architects can align the diverse concrete models with the reference models iteratively, and how they can incrementally improve the concrete and reference models until the desired level of alignment is achieved. Finally, we demonstrated the framework by applying it to an industrial case study.

The tabular representation of allocation and alignment models will enable the process of comparing models with each other, thus enabling to automate part of the alignment process. In our future work we aim to build a design support system for further assisting the business analysts and architects in the alignment process. A relevant future study in this context could be the enhancements of workflow patterns for recurring business collaboration concerns as the workflow patterns that we used were originally devised to describe centralized workflow systems of individual organisations.

ACKNOWLEDGEMENTS

The research leading to this paper received funding from the European Community's Horizon 2020 Programme (H2020/2014-2020) within the IoF2020 project under grant agreement No. 731 884. We acknowledge the European Community for the funding. We would also like to acknowledge Rob Hartog, former member of the Information Technology Group of Wageningen University, for the careful review of the paper.

REFERENCES

Alotaibi, Y. (2016). "Business process modelling challenges and solutions: a literature review." *Journal of Intelligent Manufacturing*. **27**(4): 701-723.

Angelov, S., P. Grefen and D. Greefhorst (2012). "A framework for analysis and design of software reference architectures." *Information and Software Technology*. **54**(4): 417-431.

Antunes, G., M. Bakhshandeh, J. Borbinha, J. Cardoso, S. Dadashnia, C. Di Francescomarino, M. Dragoni, P. Fettke, A. Gal, C. Ghidini, P. Hake, A. Khiat, C. Klinkmüller, E. Kuss, H. Leopold, P. Loos, C. Meilicke, T. Niesen, C. Pesquita, T. Péus, A. Schoknecht, E. Sheetrit, A. Sonntag, H. Stuckenschmidt, T. Thaler, I. Weber and M. Weidlich (2015). The Process Model Matching Contest 2015. *6th International Workshop on Enterprise Modelling and Information Systems Architectures*. J. Kolb. Innsbruck, Austria, Geellschaft für Informatik. **248:** 127-155.

APICS. (2017). "American Production and Inventory Control Society (APICS)." Retrieved January 17, 2017, from http://www.apics.org/about/overview.

Aversano, L., C. Grasso and M. Tortorella (2016). "Managing the alignment between business processes and software systems." *Information and Software Technology*. **72**: 171-188.

Barmpounakis, S., A. Kaloxylos, A. Groumas, L. Katsikas, V. Sarris, K. Dimtsa, F. Fournier, E. Antoniou, N. Alonistioti and S. Wolfert (2015). "Management and control applications in Agriculture domain via a Future Internet Business-to-Business platform." *Information Processing in Agriculture*. **2**(1): 51-63.

Barry, D. K. (2003). *Web services, service-oriented architectures, and cloud computing*, Morgan Kaufmann.

Bouguettaya, A., Q. Z. Sheng, F. Daniel and C. Pautasso (2014). RESTful Web Services: Principles, Patterns, Emerging Technologies. *Web Services Foundations*, Springer New York**:** 31-51.

Buschmann, F., K. Henney and D. C. Schmidt (2007). *Pattern-Oriented Software Architecture, Volume 4, A Pattern Language for Distributed Computing*, John wiley & sons.

Buyya, R., C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic (2009). "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation Computer Systems*. **25**(6): 599-616.

Chen, D., G. Doumeingts and F. Vernadat (2008). "Architectures for enterprise integration and interoperability: Past, present and future." *Computers in Industry*. **59**(7): 647-659.

Chen, H.-M. (2008). *Towards Service Engineering: Service Orientation and Business-IT Alignment*. Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008).

Chen, H.-M., R. Kazman and A. Garg (2005). "BITAM: An engineering-principled method for managing misalignments between business and IT architectures." *Science of Computer Programming*. **57**(1): 5-26.

Clements, P., F. Bachmann, L. Bass, D. Garlan, J. Ivers, R. Little, P. Merson, R. Nord and J. Stafford (2010). *Documenting software architectures: views and beyond*, Addison-Wesley.

Cloutier, R., G. Muller, D. Verma, R. Nilchiani, E. Hole and M. Bone (2010). "The Concept of Reference Architectures." *Systems Engineering*. **13**(1): 14-27.

Crasso, M., A. Zunino and M. Campo (2013). A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. *Innovations in Database Design, Web Applications, and Information Systems Management*, IGI Global**:** 107-138.

Cummins, F. A. (2015). BPM Meets SOA: A New Era in Business Design. *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*. J. vom Brocke and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 531-555.

Daclin, N., S. M. Daclin, V. Chapurlat and B. Vallespir (2016). "Writing and verifying interoperability requirements: Application to collaborative processes." *Computers in Industry*. **82**: 1-18.

Davenport, T. H. (2005). "The coming commoditization of processes." *Harvard Business Review*. **83**(6): 100-108.

Davenport, T. H. and J. E. Short (1990). "The New Industrial Engineering: Information Technology and Business Process Redesign." *Sloan Management Review*. **31**(4): 11-27.

Demirkan, H., R. J. Kauffman, J. A. Vayghan, H.-G. Fill, D. Karagiannis and P. P. Maglio (2008). "Service-oriented technology and management: Perspectives on research and practice for the coming decade." *Electronic Commerce Research and Applications*. **7**(4): 356-376.

Dijkman, R., M. Dumas, L. Garcia-Banuelos and R. Kaarik (2009). *Aligning Business Process Models*. 2009 IEEE International Enterprise Distributed Object Computing Conference.

Easterbrook, S., J. Singer, M.-A. Storey and D. Damian (2008). Selecting Empirical Methods for Software Engineering Research. *Guide to Advanced Empirical Software Engineering*. F. Shull, J. Singer and D. I. K. Sjøberg. London, Springer London**:** 285-311.

EC (2000). "Regulation (EC) No 1760/2000 of the European Parliament and of the Council of 17 July 2000 establishing a system for the identification and registration of bovine animals and regarding the labelling of beef and beef productsand repealing Council Regulation (EC) No 820/97." *Official Journal of the European Communities*(L 204): 1-10.

EC (2002). "Regulation (EC) No 178/2002 of the European Parliament and of the Council of 28 January 2002 laying down the general principles and requirements of food law, establishing the European Food Safety Authority and laying down procedures in matters of food safety." *Official Journal of the European Communities*. **L 31**(1): 1-24.

EC (2004). "Regulation No. 911/2004 of 29 April 2004 implementing Regulation (EC) No 1760/2000 of the European Parliament and of the Council as regards eartags, passports and holding registers." *Official Journal of the European Union*. **L 163**: 65-70.

EPCglobal (2014). EPC Information Services (EPCIS) Version 1.1 Specification. GS1 Standard Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

Erl, T. (2008). *Soa: principles of service design*, Prentice Hall Upper Saddle River.

FI-PPP. (2013). "http://www.fi-ppp.eu/." Retrieved June 4, 2013.

FIspace. (2013). "http://www.fispace.eu." Retrieved June 4, 2013.

fTRACE. (2013). "www.ftrace.de." Retrieved June 4, 2013.

Gandino, F., B. Montrucchio, M. Rebaudengo and E. R. Sanchez (2009). "On improving automation by integrating RFID in the traceability management of the agri-food sector." *Industrial Electronics, IEEE Transactions on*. **56**(7): 2357-2365.

Gregor, S. and A. R. Hevner (2013). "Positioning and Presenting Design Science Research for Maximum Impact." *MIS Quarterly*. **37**(2): 337-355.

Gruber, T. R. (1995). "Toward principles for the design of ontologies used for knowledge sharing." *International journal of human computer studies*. **43**(5): 907-928.

GS1 (2014). Core Business Vocabulary (CBV) Standard, Version 1.1, May 2014. Brussels, Belgium, GS1 AISBL.

GS1. (2017). "Global Standards One (GS1)." Retrieved January 25, 2017, from http://www.gs1.org/about.

Hammer, M. (1990). "Reengineering Work: Don't Automate, Obliterate." *Harvard Business Review*. **68**(4): 104-112.

Harmon, P. (2016). The State of Business Process Management 2016, BPTrends.

Harrington, H. J. (1991). *Business process improvement: The breakthrough strategy for total quality, productivity, and competitiveness*, McGraw-Hill Professional.

Hevner, A. and S. Chatterjee (2010). Design Science Research Frameworks. *Design Research in Information Systems*, Springer US. **22:** 23-31.

Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design Science in Information Systems Research." *MIS Quarterly*. **28**(1): 75-105.

HI-Tier. (2013). "Herkunftssicherungs- und Informationssystem für Tiere." Retrieved June 4, 2013.

Hinkelmann, K., A. Gerber, D. Karagiannis, B. Thoenssen, A. van der Merwe and R. Woitsch (2016). "A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology." *Computers in Industry*. **79**: 77-86.

Hoch, R., H. Kaindl, R. Popp and C. Zeidler (2016). *Aligning Architectures of Business and Software: Software Driven by Business Process Models and Its User Interface*. 2016 49th Hawaii International Conference on System Sciences (HICSS).

ISO/IEC (2013). Information technology - Object Management Group Business Process Model and Notation. ISO/IEC 19510:2013.

ISO/IEC/IEEE (2011). Systems and software engineering -- Architecture description. *ISO/IEC/IEEE Standard 42010:2011*.

Kaloxylos, A., J. Wolfert, T. Verwaart, C. M. Terol, C. Brewster, R. Robbemond and H. Sundmaker (2013). "The Use of Future Internet Technologies in the Agriculture and Food Sectors: Integrating the Supply Chain." *Procedia Technology*. **8**: 51-60.

Kassahun, A., R. J. M. Hartog, T. Sadowski, H. Scholten, T. Bartram, S. Wolfert and A. J. M. Beulens (2014). "Enabling chain-wide transparency in meat supply chains based on the EPCIS global standard and cloud-based services." *Computers and Electronics in Agriculture*. **109**(0): 179-190.

Kassahun, A., R. J. M. Hartog and B. Tekinerdogan (2016). "Realizing chain-wide transparency in meat supply chains based on global standards and a reference architecture." *Computers and Electronics in Agriculture*. **123**: 275-291.

Kürschner, C., C. Condea, O. Kasten and F. Thiesse (2008). Discovery Service Design in the EPCglobal Network. *The Internet of Things: First International Conference, IOT 2008, Zurich, Switzerland, March 26-28, 2008. Proceedings*. C. Floerkemeier, M. Langheinrich, E. Fleisch, F. Mattern and S. E. Sarma. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 19-34.

Kywe, S. M., J. Shi, Y. Li and R. Kailash (2012). "Evaluation of Different Electronic Product Code Discovery Service Models." *Advances in Internet of Things*. **Vol.02No.02**: 10.

Laplante, P. A. (2013). *Requirements engineering for software and systems*, CRC Press.

Liu, C., Q. Li and X. Zhao (2009). "Challenges and opportunities in collaborative business process management: Overview of recent advances and introduction to the special issue." *Information Systems Frontiers*. **11**(3): 201-209.

Lorenz, M., A. Zeier, H. Plattner, J. Müller and M.-P. Schapranow (2011). Discovery services in the EPC network. *Designing and Deploying RFID Applications*. C. Turcu, InTech.

Mentzer, J. T., W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith and Z. G. Zacharia (2001). "DEFINING SUPPLY CHAIN MANAGEMENT." *Journal of Business Logistics*. **22**(2): 1-25.

Mitra, N. (2003). "Soap version 1.2 part 0: Primer." *W3C Recommendation*. **24**.

Moe, T. (1998). "Perspectives on traceability in food manufacture." *Trends in Food Science & Technology*. **9**(5): 211-214.

Murphy, G. C., D. Notkin and K. J. Sullivan (2001). "Software reflexion models: bridging the gap between design and implementation." *Software Engineering, IEEE Transactions on*. **27**(4): 364-380.

Mynetfair. (2013). "www.mynetfair.com " Retrieved June 4, 2013.

Newcomer, E. and G. Lomow (2004). *Understanding SOA with Web Services*, Addison-Wesley Professional.

Nuseibeh, B. and S. Easterbrook (2000). Requirements engineering: a roadmap. *Proceedings of the Conference on The Future of Software Engineering*. Limerick, Ireland, ACM.

OASIS (2004). UDDI Version 3.0.2. **uddi_v3**.

OASIS (2006). Reference model for service oriented architecture 1.0. *OASIS Standard*. **12**.

OASIS. (2017). "Organization for the Advancement of Structured Information Standards (OASIS)." Retrieved January 7, 2017, from https://www.oasis-open.org/.

OMG. (2017). "Object Managment Group (OMG)." Retrieved January 25, 2017 from http://www.omg.org/.

Papazoglou, M. P., P. Traverso, S. Dustdar and F. Leymann (2008). "Service-Oriented Computing: A Research Roadmap." *International Journal of Cooperative Information Systems*. **17**(02): 223-255.

Peltz, C. (2003). Web Services Orchestration and Choreography. **36:** 46-52.

Pradabwong, J. (2017). "Business process management and supply chain collaboration: effects on performance and competitiveness." *Supply Chain Management: An International Journal*. **22**(2): 107-121.

QS. (2013). "Qualität und Sicherheit (QS)." Retrieved June 4, 2013, from https://www.q-s.de/en/.

Rosemann, M. and J. vom Brocke (2015). The Six Core Elements of Business Process Management. *Handbook on Business Process Management 1: Introduction, Methods, and Information Systems*. J. vom Brocke and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 105-122.

Runeson, P. and M. Höst (2008). "Guidelines for conducting and reporting case study research in software engineering." *Empirical Software Engineering*. **14**(2): 131.

Runeson, P., M. Host, A. Rainer and B. Regnell (2012). *Case study research in software engineering: Guidelines and examples*, John Wiley & Sons.

Russell, N., W. M. van der Aalst and N. Mulyar (2006). "Workflow Control-Flow Patterns: A Revised View." *BPM Center Report BPM-06-22*.

Sadiq, S., G. Governatori and K. Namiri (2007). Modeling Control Objectives for Business Process Compliance. *Business Process Management: 5th International Conference, BPM 2007, Brisbane, Australia, September 24-28, 2007. Proceedings*. G. Alonso, P. Dadam and M. Rosemann. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 149-164.

Simon, H. A. (1996). *The sciences of the Artificial*. Cambridge, MA [etc.], The MIT Press.

Stirna, J. and J. Zdravkovic (2015). "Interview with sladjan maras on "challenges and needs in enterprise modeling"." *Business & Information Systems Engineering*. **57**(1): 79-81.

The Open Group (2011). *TOGAF Version 9.1*. Zaltbommel, Van Haren Publishing.

The Open Group. (2017). "The Open Group." Retrieved January 25, 2017, from http://www.opengroup.org/.

van der Aalst, W. M. P. and A. ter Hofstede. (2011). "Workflow Patterns. http://www.workflowpatterns.com/." Retrieved December 23, 2015.

Van der Aalst, W. M. P., A. H. M. Ter Hofstede, B. Kiepuszewski and A. P. Barros (2003). "Workflow Patterns." *Distributed and Parallel Databases*. **14**(1): 5-51.

Verdouw, C., A. Beulens and S. Wolfert (2014). *Towards Software Mass Customization for Business Collaboration*. Global Conference (SRII), 2014 Annual SRII.

W3C (2007). Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language, W3C. **REC-wsdl20-20070626**.

Weidlich, M., J. Mendling and M. Weske (2011). "Efficient Consistency Measurement Based on Behavioral Profiles of Process Models." *IEEE Transactions on Software Engineering*. **37**(3): 410-429.

Wieringa, R. J., H. M. Blanken, M. M. Fokkinga and P. W. P. J. Grefen (2003). Aligning Application Architecture to the Business Context. *Advanced Information Systems Engineering: 15th International Conference, CAiSE 2003 Klagenfurt/Velden, Austria, June 16–20, 2003 Proceedings*. J. Eder and M. Missikoff. Berlin, Heidelberg, Springer Berlin Heidelberg**:** 209-225.

Wolfert, J., C. N. Verdouw, C. M. Verloop and A. J. M. Beulens (2010). "Organizing information integration in agri-food—A method based on a service-oriented architecture and living lab approach." *Computers and Electronics in Agriculture*. **70**(2): 389-405.

Yin, R. K. (2003). "Case Study Research: Design and Methods." *SAGE*(181): 28.

Zachman, J. A. (1987). "A framework for information systems architecture." *IBM Systems Journal*. **26**(3): 276-292.

Zhao, X., C. Liu, Y. Yang and W. Sadiq (2009). "Aligning Collaborative Business Processes—An Organization-Oriented Perspective." *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*. **39**(6): 1152-1164.