Towards Dependency-based Alignment for Collaborative Businesses

Novica Zarvić, Michael Fellmann, Oliver Thomas

Institut für Informationsmanagement und Unternehmensführung
Universität Osnabrück
Katharinenstr. 3
49074 Osnabrück, Germany
{novica.zarvic, michael.fellmann, oliver.thomas}@uni-osnabrueck.de

Abstract: Collaborative Businesses are often interpreted as the new organisational form of the 21st century. Such organisational forms exist to satisfy complex needs, which one company could usually not satisfy on its own. This means that the participating companies in Collaborative Businesses are to a certain degree dependent on each other. From a business-IT alignment perspective, Collaborative Businesses require a different treatment than traditional organisational forms. The approach presented in this paper builds upon the aforementioned dependency aspect. We place the different inter-organisational dependencies, which are relevant in this context, into a layered alignment framework and discuss the conceivable alignment perspectives. Then we analyse the dependency impact on the life cycle phases of Collaborative Businesses. Finally, we exemplify how to operationally use our dependency-based alignment framework and how dependency-based alignment can be supported with Description Logics.

1 Introduction

The emerging field of Collaborative (Business) Network research[1] has the aim to advance theory and practice that is needed to deal with the new organisational form of Collaborative Businesses [CA04]. It is sometimes even considered to represent a new scientific discipline, on which currently a large number of research projects are carried out worldwide [CA05]. The most apparent trait of Collaborative Businesses is that two or more independent companies are working together towards a common goal, which one company could not achieve on its own [GA03], which consequently means that the companies are becoming dependent on each other. This view is in line with Kanter's statement that interdependence is representing one of the main characteristics of intercompany relationships, where business partners need each other, because of possessing complementary assets and skills [Kan94].

[1] See http://www.socolnet.org/ (SoColNet – Society of Collaborative Networks) for up-to-date information on activities performed in the Collaborative Business research field.

Therefore, the management of dependency relations becomes a crucial and complex task for the success of any type of Collaborative Business. Dependency relations can be found and respectively observed at any describable enterprise architecture level (cf. Section 2.2). Such a dependency relation (e.g. included in a specific model) at one enterprise architecture level might have some impact on other dependency relations at other levels. Consequently, a change in a model at one certain level might cause changes in other models at other levels, which makes the management of dependency relations an alignment issue. Business-IT alignment is a difficult task even within a single company, but in a Collaborative Business constellation, where multiple businesses collaborate with each other, it gains complexity [ZWE08]. This paper aims to contribute here by presenting a dependency-based approach to the management and alignment of dependencies in the area of Collaborative Businesses. For being able to do so, following research questions need to be answered:

> ***Q1:*** *How to structure the different dependency types from literature systematically in order to ensure a holistic view and how many alignment perspectives are conceivable?*
>
> ***Q2:*** *Which impact do the identified dependency types have on the life cycle phases of a Collaborative Business?*
>
> ***Q3:*** *How alignment needs to be performed and ensured on the basis of inter-organisational dependencies and how can it formally be supported?*

The paper is structured as follows. In Section 2 we give a compact overview on the state-of-the-art in Business-IT Alignment, before we present the dependency-based alignment framework. This answers ***Q1***. In Section 3, we discuss the impact of inter-organisational dependencies on the life cycle phases of a Collaborative Business, which answers ***Q2***. Section 4 exemplifies how to use the proposed alignment tool, which gives an answer to the first part of ***Q3***. In Section 5 it is shown how support can be provided to the alignment activities by means of Description Logics, which answers the second part of ***Q3***. Section 6 finally concludes the paper.

2 Towards a Dependency-based Alignment Framework

2.1 State-of-the-Art in Business-IT Alignment

Business-IT alignment presents now for more than two decades a top concern for both academia and practice [CR07]. Among the first approaches of alignment between business and IT was provided by using diverse structured Information Systems planning methodologies [LM89] such as Strategic Data-Planning [Mar82], Business Systems Planning by IBM [Zac82] or Information Engineering [Mar89, Fin89]. All these methodologies have in common that they all follow a top-down approach and neglect legacy systems and were designed for single companies only. Furthermore, these approaches are planning the system landscape of a company completely from scratch.

Business-IT alignment can be categorised into *strategic alignment* and *operational alignment*. Strategic alignment is more of descriptive nature and tells *what* has to be done, whereas operational alignment is more of constructivist nature and tells *how* to do things. The notion of *alignment* itself was introduced and shaped slightly after the introduction of the structured Information Systems planning methodologies by Henderson & Venkatraman [HV93]. They described alignment from a strategic point of view, which is the predominant category that can be found and which is mostly covered by the Information Systems community in the Anglo-American area. Operational alignment on the other side is currently mainly covered by the Information Systems Engineering community in the European area, like e.g. [BT08, Wie08], except the early Information Systems planning methodologies that were mentioned above. However, literature on operational alignment still seems to be underrepresented. These methodologies, mainly stemming from the 1970s and 1980s, neither considered bottom-up nor out-of-the-middle alignment activities. This deficit was first addressed by Enterprise Architecture Frameworks (EAFs), like for instance the Zachman Framework [Zac87, SZ92]. Such EAFs are often structured by means of layers and are having the aim to organise company-wide knowledge and make complexity more controllable. They also allow for top-down, bottom-up, and out-of-the middle perspectives on Business-IT alignment. Nevertheless, layered EAFs, like e.g. the Zachman Framework, mainly provide company-wide documentation structures, but no methodology of how to implement objects [Zac08], and hence do not provide sufficient support of performing operational Business-IT alignment. Note that nowadays Business-IT alignment is often viewed as complementary to IT-Governance and IT-Compliance respectively [WR04].

We do not claim to have presented here a complete state-of-the-art on Business IT alignment, but we gave a compact chronological overview on main activities performed in this research field in the past. For a more thorough overview refer to [CR07].

2.2 Inter-organisational Dependencies

Cox et al. claim that analysing dependencies between entities is a very important part of modelling. Further they state that "much of the present literature takes the definition of dependency for granted and where definitions are occasionally given, they widely vary" [CDS01, p. 118]. Therefore many publications address dependencies in widely varying ways and apply them to different contexts. We searched the literature for dependencies in an inter-organisational context and found following application areas that need later on to be integrated into an adequate structure (cf. ***Q1***):

Business and organisational dependencies: Emerson [Eme62] investigated complex community relations and proposed a simple theory of *power-dependence* relations by studying power asymmetries in collaborative structures. *Interdependencies*, as proposed in organisational theory by Thompson [Tho67], initially took an intra-organisational view to describe three recurring interdependence patterns, which are pooled, sequential and reciprocal interdependence. Kumar and van Dissel discussed later these configuration patterns – representing organisational structures – for inter-organisational settings [KD96].

Product and Service dependencies: Dependencies between product parts have thoroughly been investigated by means of Gozinto-graphs [Alt03], which essentially show the product structure including the number of all required product parts. *Resource-dependency theory* assumes that one organisation cannot possess or produce all required resources alone. Hence, this gap should be filled by assessing resources that are managed by others [PS78]. Such resource-resource dependencies are also described by Crowston [Cro94] within the field of coordination theory. Dependencies have also been investigated with respect to services. The so-called *molecular models*, which were introduced by Shostack [Sho93], show relations between services and service elements. Winkler and Schill [WS09] have also investigated *service dependencies* in service compositions in the logistics domain. Ludwig and Franczyk [LF08] describe dependencies and relations between Service Level Agreement (SLA) elements in atomic and composite services and provide with *COSMA* (Composite SLA Management) a solution for managing these dependencies. A similar approach is provided by Bodenstaff [Bod10] with the *MoDe4SLA* approach (Monitoring Dependency relations for SLAs).

Process, function and task dependencies: Coordination theory provided with *task-task dependencies* and *task-resource dependencies* important input for managing dependencies at process level [Cro94]. Another approach for managing dependency relations in coordination models on the basis of business models is provided by Bodenstaff [Bod10]. Nevertheless, it can be said that the management of dependencies in processes or workflows is an important aspect of BPM (Business Process Management), where we have to deal especially with time-based dependencies [CDS01] in order to manage the order of process steps.

ICT system and application dependencies: ICT systems and applications are representing the underlying infrastructure for enabling the process level described above. Consequently, for running inter-organisational processes properly there exist also dependencies of the underlying ICT infrastructure elements [SP08]. So-called communication diagrams show communication channels between ICT systems and applications respectively [Lan09]. Here an application might be dependent on the data stored in an information system. Another approach that partially tackles dependencies at ICT level is provided by Ball-Rokeach [Bal85], who introduced the *media-system dependency*. However, this approach is not limited to the ICT world and elements only, but builds on and is linked to the social world.

Other dependencies: *Strategic dependency* situations are widely discussed in literature [PC06]. However, these strategic dependency situations refer to dependencies (e.g. actor dependency, resource dependency, task dependency [YM93]) that can be situated in one of the dependency categories described above.

2.3 Definition of the Dependency-based Alignment Framework

The starting point for defining the envisioned dependency management tool was to decide how to organise and manage the complexity of the multitude of inter-organisational dependencies found in the literature. We started from the notions of *system* and *architecture of a system.* A system is a basic and fundamental concept in systems engineering, which enables us to reason about the world, or even parts of it. We consider a Collaborative Business to be such a system. The systems' architecture on the other side is the structure of components and their properties including the interactions between these components that realise system-level properties. The components deliver services to the composite system, meaning that component's services can not be achieved without using the services of other components, resulting in a service-provision relationship. This led us to a layered structure, where entities from a lower level provide service and thus realise, support [Wie03] or can at least be used to justify the existence of entities at higher levels. Such layered architecture styles are, as already mentioned, well-known in the context of EAFs. Take for instance the Zachman Framework with its different layers, where business processes are supported and realised by the underlying software applications, which in turn are supported and realised by the underlying hardware systems [Zac82, Zac87].

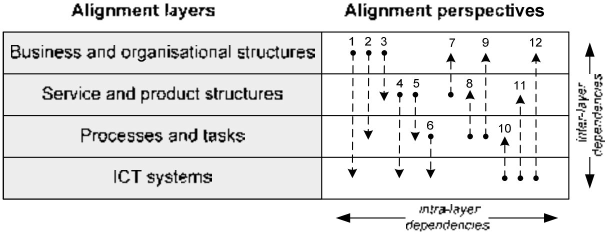


Figure 1: The Dependency-based Alignment Framework

Figure 1 presents our dependency-based alignment framework, where a layered architectural style was used for constructing it. The number of alignment perspectives is *n(n-1)*, where *n* represents the number of layers. This amounts in our case to 12 different alignment perspectives. The number of layers in our dependency-based alignment framework is of course freely adjustable; layers may be merged or new layers may be added. Consequently, this would change the number of possible alignment perspectives according to the simple formula stated above. This answers ***Q1***. Note that we do not call our framework a Business-IT Alignment framework, but simply an alignment framework. Even though the notion Business-IT alignment became widely accepted, it seems to be a bit too narrow, because it limits terminologically alignment to the two terms.

In the context of our work and in line with [CDS01], we define entities in the context of modelling languages that depend on other entities *dependents*, while entities on which other entities are dependent on are called *antecedents*. In complex environments, such as Collaborative Businesses, an entity can often be both, a dependent and an antecedent. Further, one could make the distinction between *intra-layer dependencies and inter-layer dependencies*. Intra-layer dependencies are dependencies between entities that are observable within one layer, whereas inter-layer dependencies are dependency relations between entities in different layers.

3 Dependency Impact during the Collaborative Business Life Cycle

Thoben and Jagdev describe the formal duration of collaborations between different companies or organisations by what they call the life cycle of an enterprise network [TJ01]. They distinguish thereby four different life cycle phases: a *preparation* phase (preparing, first sourcing of partners, etc.), a *formation/setting up* phase (final partner selection, legal issues, contracts, etc.), an *operation* phase (day-to-day management of the network), and a *dissolution* phase (decomposition of the network).

Seifert builds on this work and compares the duration of different kinds of Collaborative Businesses. He thereby considers traditional supply chains as stable collaborations and virtual organisations as dynamic collaborations respectively. The main difference between these two is that the operation phase lasts much longer in traditional stable collaborations than in virtual organisations, which usually have only a short temporary character. However, Seifert distinguishes in his work just three phases, where the preparation and formation phase are merged into a single phase, named the *initiation* phase. Within the initiation phase he distinguishes three major steps, the *initiation*, the *partner selection* and finally the *agreement*. During the second step, the partner selection, the consortium is built [Sei09]. Next to these two life cycle definitions there exists also the Collaborative Network life cycle by Camarinha-Matos and Afsarmanesh [CA07], who have also defined four life cycle phases like Thoben and Jagdev [TJ01], but divide the last phase into two distinct phase possibilities.

Considered dependency	**Preparation phase**	**Formation phase**	**Operation phase**	**Dissolution phase**	**Metamor-phosis phase**
Organisational interdependencies	⊙	●	●	○	●
Resource-resource dependencies	⊙	●	●	⊙	●
Task-task dependencies	○	⊙	●	○	⊙
System dependencies	○	⊙	●	○	⊙
	Legend: ● high impact	⊙ medium impact	○ low impact		

Table 1: Impact of Inter-organisational dependencies on Collaborative Business Life Cycle Phases

In our analysis of the impact of inter-organisational dependencies (cf. ***Q2***) we considered the life cycles with the highest number of phases. These are, as already mentioned, in the case of [TJ01] four fix phases and at [CA07] also four phases, where only the first three are fix, but the fourth phase can either represent the dissolution or a metamorphosis of the Collaborative Business. Inter-organisational dependencies are present throughout the life cycle of Collaborative Businesses. During the preparation phase only organisational and product/service related dependencies have high impact. Then, a high impact of inter-organisational dependencies can be observed during both the formation and operation phases. The lowest impact can be observed in the dissolution phase, which is also affirmed by the literature that we have reviewed. In case there is a metamorphosis at the end of the life cycle, the impact of the inter-organisational dependencies is again high, because the dependencies gain importance during this transformation step, which is similar to the formation phase. The dependency impact analysis in Table 1 is based on studying the activities within the respective phases and valuating the importance of the discussed inter-organisational dependencies for each of the phases.

4 Application of the Dependency-based Alignment Framework

In the following we exemplify the operational usage of our dependency-based alignment framework on the basis of a recent example from literature [ZST10]. This example is to be placed within the formation phase of a Collaborative Business (cf. Section 3). We will thereby look especially at two different dependencies, namely inter-organisational dependencies and resource-resource dependencies, because these have been appointed a high impact during the formation of a Collaborative Business (cf. Table 1).

4.1 Example case description

Consider a small manufacturing firm for office equipment that mainly offers the customers a service to design and configure an individual ballpoint pen via the internet, collects related components desired by the customer from network partners and assembles/distributes the final product. Thereby, a recently performed market analysis came to the result that there is a business opportunity for customised ballpoint pens with high quality refills on the targeted market. Such a product is currently not in the portfolio of the office equipment assembler, which leads to the decision to expand the company's product range. The end consumers are enabled by means of a web service to configure the ballpoint pens by specifying product-specific properties, like the colour of the ballpoint pen, material, an individual logo or even the colour of the high quality refills of the ballpoint pen.

The design team has figured out the product-specific resources are needed in order to be able to fulfil the task to assemble and individually customize the ballpoint pens. These are (i) plastic parts, (ii) steel parts, and (iii) high quality refills. Note that there exist, next to the product-specific resources also resources that are needed for the production, but do not make up the product themselves. Take as example the diverse tools and machines that are involved in such processes.

4.2 Performing alignment on the basis of inter-organisational dependencies

In this subsection we propose a simple and pragmatic methodology for performing alignment within our dependency-based alignment framework. Our methodological approach – providing an answer for the first part of ***Q3*** – consists of three major steps, which are:

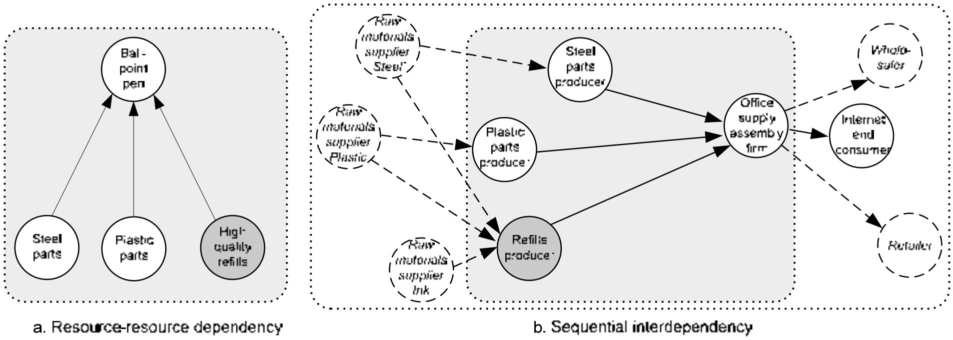
1. **Describing dependency changes in one layer:** It is observed that a change of a dependency relation within one layer is occurring. This might be the result of an outsourcing decision, a new product design, expansion of service functionalities and the like. This change is described by means of a graphical model (or in Section 5 by means of Description Logics syntax).
2. **Consistency checking:** In the second step it has to be checked whether the change has some impact on current practices. This means for the case at hand that it must be checked whether the new product is still producible. In case it is not anymore producible, this entails that some changes at global level must occur, thus inter-organisational dependencies must be aligned with each other. Remember that an observed change in a dependency relation in one layer might cause a change in dependency relations in other layers, but this is not obligatory.
3. **Detecting inter-layer dependencies:** The final step gives an indication of what kinds of changes are required in the dependency relations in the layers under consideration.

4.3 From Product Structure to Collaborative Business Structure

Describing dependency changes in one layer: The trend of reducing in-house production depth [Qui99] has also reached the small manufacturing firm for office equipment, which is just specialised on assembling products, rather than producing every product part in-house.

We start looking at the product structure of the envisioned ballpoint pen (more precisely at the product specific resources) from a resource-resource dependency perspective [Cro94]. The Gozinto-graph simply shows that the final product, namely the ballpoint pen, is dependent on the different parts it is made from. A representation by means of a Gozinto-graph is shown in Figure 2a.

Consistency checking: The dependencies on plastic parts and steel parts already existed at older products in the portfolio of the small manufacturing firm for office equipment, but the dependence on the product component high quality refills is new, because the company has only assembled products with low quality refills or pen cartridges before. Hence the resource-resource dependency of the new ballpoint pen changed. This in turn means that the envisioned product can not be produced and assembled with the existing dependencies.



a. Resource-resource dependency

b. Sequential interdependency

Figure 2: a) Resource-resource dependency at product level requires change in b) sequential interdependency at organisational level

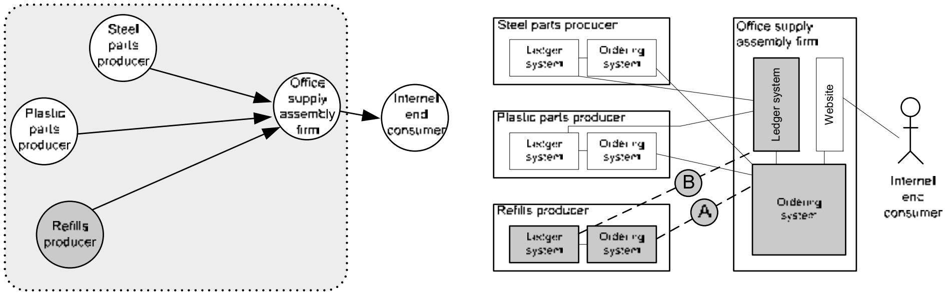
Detecting inter-layer dependencies: We take here alignment perspective 7 from our alignment framework into account, because of the high impact identified in our life cycle analysis before. The question arises whether this dependency change has some impact on the layer "business and organizational structures". With respect to the plastic parts and for the steel parts suppliers already exist. These are long term suppliers that already provide similar parts needed for the product range of the assembly firm. In order to enable the new product structure, the office supply assembly firm needs – as the lead partner of the Collaborative Business constellation – to form an additional partnership with a supplier that can supply the missing component. This of course changes the current constellation of the sequential interdependency in the overall supply chain.

This example shows that alignment, when managing a Collaborative Business holistically, is not limited only to Business and IT. A change in the product structure might require a change in the organisational structure of the Collaborative Business, which as a consequence needs to be aligned, as shown in Figure 2b.

4.4 From Collaborative Business Structure to ICT Systems landscape

Describing dependency changes in one layer: The alignment perspective 7 led to a change in the sequential inter-organisational interdependence, because a new partner for refills supply was added to the chain. This change in the collaboration structure is shown in Figures 2b and 3a respectively. However, this change might again cause changes in dependency relations at other levels.

Consistency checking: Our lead partner in the collaboration, namely the assembly firm for office supply, will only get into collaboration with a new refills supplier if the main processes in day-to-day business are supported by means of ICT. This includes that the implemented applications realise these processes. Following applications are currently implemented at each partner: a *ledger system* for financial administration and an *ordering system* for ordering needed resources from the suppliers. This means, as far as the systems of the new partner are not integrated into the collaborative systems landscape, that the current systems do not support the aforementioned two processes.



a. Sequential interdependency

b. Application dependencies

Figure 3: a) Sequential interdependency relation at organisational level requires changes in b) application dependencies at ICT level

Detecting inter-layer dependencies: We take here alignment perspective 1 from our alignment framework into account, in order to exemplify also alignment on the ICT part within our framework. In Figure 3b, a classical communication diagram, the applications that are affected by the sequential interdependency at organizational level are highlighted. Equally the missing communication channels (A) and (B) need to be established to realize the required application dependencies between the applications at the refills producer and at the office supply assembly firm. Hence, the communication diagram shows how to align the dependencies between ICT systems according to changes in the collaboration structure.

5 Supporting Dependency-based Alignment with Description Logics

The aforementioned application of dependency-based alignment can be supported by means of ICT in various ways. In the following we present an approach based on Description Logics, which also serves as a prototypical proof-of-concept and provides an answer to the second part of ***Q3***.

We show a) how to indirectly describe dependencies and dependency changes by modifying the descriptions in a single layer, b) how to perform consistency checks based on a global view spanning different alignment layers and c) how to detect dependencies between different alignment layers arising due to e.g. changes within a single layer. Our main objective for using Description Logics is to automate both the consistency checks and the detection of dependencies spanning different layers. Most Description Logics are a fragment of First Order Logic, a prominent example is the description logic SHOIN(D) underlying the well-standardized Web Ontology Language (OWL) (w3.org/2007/OWL). As we do not advocate for a specific type of Description Logic, we hence present our examples in the more general DL-syntax. For subsumption, the symbol $\sqsubseteq$ is used, logical operators are written as $\sqcap$ (AND) and $\sqcup$ (OR), equivalence is denoted by the symbol $\equiv$ and quantifications are represented by $\exists$ (exists) and $\forall$ (all).

5.1 Indirect Description of Dependencies

In the following, we present some formal descriptions of the product layer. As an example, we describe three components of a product which are exclusively assigned to suppliers being external organisational units, although in the case of component C the supplier is not (yet) known being the reason why we assign an unknown supplier as the range of the property `hasSupplier`.

$$\begin{aligned}\mathtt{ComponentA} \sqsubseteq\ & \exists\mathtt{hasSupplier.ExternalSupplier} \\ & \sqcap\ \forall\mathtt{hasSupplier.ExternalSupplier}\end{aligned}$$

$$\begin{aligned}\mathtt{ComponentB} \sqsubseteq\ & \exists\mathtt{hasSupplier.ExternalSupplier} \\ & \sqcap\ \forall\mathtt{hasSupplier.ExternalSupplier}\end{aligned}$$

$$\mathtt{ComponentC} \sqsubseteq \forall\mathtt{hasSupplier.UnknownSupplier}$$

Also, we describe a product composed exclusively of these formerly described components with at least one component of each going into the final product.

$$\begin{aligned}\mathtt{MyProduct} \sqsubseteq\ & \exists\mathtt{hasPart.ComponentA} \sqcap \exists\mathtt{hasPart.ComponentB} \\ & \sqcap\ \exists\mathtt{hasPart.ComponentC} \\ & \sqcap\ \forall\mathtt{hasPart.(ComponentA} \sqcup \mathtt{ComponentB} \\ & \qquad\qquad \sqcup\ \mathtt{ComponentC)}\end{aligned}$$

Now, we have described a little fragment of the product layer. With these descriptions, we also have indirectly described a dependency between the product layer and the organisational layer as we will see later on in the section “detection of dependencies”.

5.2 Consistency checking

In order to check the consistency of the descriptions in the product layer against a set of global constraints, we represent those constraints as class descriptions. As an example, we constrain our products to be composed exclusively of parts for which suppliers are known. To check if all products adhere to this constraint, we classify products *not* adhering to this constraint as unfabricable products. This is done by a class description stating that each product for which a part exists that has an unknown supplier is an unfabricable product.

$$\mathtt{UnFabricableProduct} \equiv \exists\mathtt{hasPart.(}\exists\mathtt{hasSupplier.UnknownSupplier)}$$

This description of a defined class (in DL-syntax shown by the symbol ≡) can be used to determine automatically the set of things adhering to this definition by using an inference engine. In this way, we easily can detect if our product falls into the class of unfabricable products by running the inference engine (i.e. by classifying the taxonomy) and observing the changes in the class hierarchy. As the definition of MyProduct states that this product also contains ComponentC – a component for which no supplier is known – we obtain the result that MyProduct is in fact subsumed by UnFabricableProduct.

$$\texttt{MyProduct} \sqsubseteq \texttt{UnFabricableProduct}$$

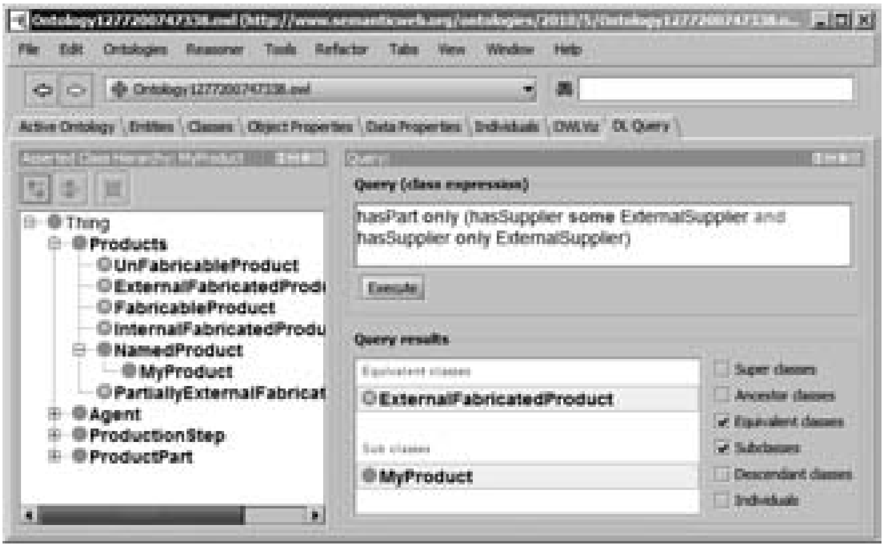
After changing the description of ComponentC e.g. in such a way that it is also supplied by at least one external supplier and only external suppliers, it is no longer subsumed by UnFabricableProduct.

5.3 Detecting dependencies

In order to detect dependencies between the product layer and the organisational layer, we define additional classes corresponding to each dependency. For example, we define a class ExternalFabricatedProduct containing all products made of parts which are produced by at least one external supplier and exclusively by external suppliers.

$$\begin{aligned}&\texttt{ExternalFabricatedProduct} \equiv \\ &\quad \forall \texttt{hasPart.}(\exists \texttt{hasSupplier.ExternalSupplier} \\ &\qquad\qquad \sqcap \forall \texttt{hasSupplier.ExternalSupplier})\end{aligned}$$

Products which are subsumed by this class – which again can be computed automatically using an inference engine – represent a dependency between the product and the organisational layer. This dependency is constituted by the fact that products subsumed by this class are contingent upon external organisational units. Each newly described product type hence can be checked in regard to this dependency, and if the product is subsumed by this class, then a dependency to external parties would be established by producing this product. We can interpret such descriptions of defined classes also as a query since an inference engine is capable of automatically computing all members of the class. This usage of class descriptions as a query is demonstrated by the Protégé ontology editor (protege.stanford.edu) providing a tab for description logic queries (DL Query). If we use the above definition of ExternalFabricatedProduct as a Query, we precisely get the result that our product MyProduct is subsumed by ExternalFabricatedProduct, i.e. it is assembled by components that were firstly externally fabricated and then supplied to our office supply assembly firm. Thus, a dependency to external organisational units is established (note that the DL-syntax has to be translated to the syntax used by Protégé which essentially replaces mathematical symbols by English words such as only, some, and, or).



Ontology1277200747338.owl

- Thing
  - Products
    - UnFabricableProduct
    - ExternalFabricatedProd
    - FabricableProduct
    - InternalFabricatedProdu
    - NamedProduct
      - MyProduct
    - PartiallyExternalFabricat
  - Agent
  - ProductionStep
  - ProductPart

Query (class expression)

hasPart only (hasSupplier some ExternalSupplier and hasSupplier only ExternalSupplier)

Query results

Equivalent classes

ExternalFabricatedProduct

Sub classes

MyProduct

□ Super classes
□ Ancestor classes
☑ Equivalent classes
☑ Subclasses
□ Descendant classes
□ Individuals

Figure 4: DL-Query inside the Protégé ontology editor

6 Summary and Future Outlook

Business-IT alignment is an actual topic in multiple research fields, such as Information Systems, Information Systems Engineering, Collaborative Businesses, etc. In this paper we proposed a novel framework for performing alignment in Collaborative Businesses. The framework is based on inter-organisational dependencies. The initially stated goals by means of research questions have been answered sufficiently. In detail, we have chosen a layered architecture style for being able to systematically arrange the different dependency types from literature. Further we have stated all conceivable alignment perspectives within the framework, which enable a top-down, bottom-up or out-of-the-middle alignment. This answered ***Q1***. Then we have answered ***Q2*** in Table 1, by analysing the dependency impact on the life cycle phases of a Collaborative Business. In order to make the alignment framework an operational and not only a strategic tool, we have defined a pragmatic method that was presented on the basis of an example in Section 4, which was followed by a discussion and a prototypical proof-of-concept of how Description Logics can support operational alignment within the dependency-based alignment framework. This finally gave an answer to ***Q3***.

Future work will imply a further elaboration on the intra- and inter-layer dependencies, which will be accompanied by the definition of crisp consistency rules. It should be noted that the presented framework supports the alignment process by indicating in which layer changes of the dependency constellation are likely to occur and have an impact. With respect to the example in this paper, we identified e.g. that a partner is needed, which in turn means that the Collaborative Business structure needs to be aligned to the new product structure, but we did not identify whom exactly to choose. For the activity of partner selection specific methods have been developed.

References

[Alt03] Altrogge, G.: Stücklisten und lineare Produktionsstrukturen. Verlag Dr. Kovač, Hamburg, 2003.

[Bal85] Ball-Rokeach, S.J.: The origins of individual media-system dependency: A sociological framework. In Communication Research, Vol. 12, No. 4, 1985; pp. 485-510.

[Bod10] Bodenstaff, L.: Managing Dependency Relations in Inter-Organizational Models. PhD thesis, University of Twente, The Netherlands, 2010.

[BT08] Blankenhorn, H.; Thamm, J.E.: Business-IT Alignment – Aufbau und Operationalisierung der IT-Strategie. In Information Management & Consulting, Vol. 23, No. 1, 2008; pp. 9-16.

[CA04] Camarinha-Matos, L.; Afsarmanesh, H.: The emerging discipline of collaborative networks. In Proc. of 5th Working Conference on Virtual Enterprises, 23-26 Aug 2004, Toulouse, France, Kluwer Academic Publishers, 2004; pp. 3-16.

[CA05] Camarinha-Matos, L.; Afsarmanesh, H.: Collaborative networks: a new scientific discipline. In Journal of Intelligent Manufacturing, Vol. 16, No. 4-5, 2005; pp. 439-452.

[CA07] Camarinha-Matos, L.; Afsarmanesh, H.: A comprehensive modeling framework for collaborative networked organizations. In Journal of Intelligent Manufacturing, Vol. 18, No. 5, 2007; pp. 529-542.

[CDS01] Cox, L.; Delugach, H.S.; Skipper, D.: Dependency Analysis Using Conceptual Graphs. In Proc. of 9th Int. Conference on Conceptual Structures (ICCS2001), 2001; pp. 117-130.

[CR07] Chan, Y.; Reich, B.: IT alignment: what have we learned?. In Journal of Information Technology, Vol. 22, No. 4, 2007; pp. 297-315.

[Cro94] Crowston, K.: A taxonomy of organizational dependencies and coordination mechanisms. Working paper (Sloan School of Management), MIT, Cambridge, Massachusetts, TR #174, 1994.

[Eme62] Emerson, R.M.: Power-Dependence Relations. In American Sociological Review, Vol. 27, No. 1, 1962; pp. 31-41.

[Fin89] Finkelstein, C.: An Introduction to Information Engineering – From Strategic Planning to Information Systems. Addison-Wesley, Sydney, 1989.

[GA03] Gordijn, J.; Akkermans, H.: Value-based requirements engineering. In Requirements Engineering, Vol. 8, No. 2, 2003; pp. 114-134.

[HV93] Henderson, J.; Venkatraman, N.: Strategic Alignment: Leveraging information technology for transforming organizations. In IBM Systems Journal, Vol. 32, No. 1, 1993; pp. 472-484.

[Kan94] Kanter, R.: Collaborative advantage: The art of alliances. In Harvard Business Review, Vol. 72, No. 4, 1994; pp. 96-108.

[KD96] Kumar, K.; van Dissel, H.G.: Sustainable collaboration: managing conflict and cooperation in interorganizational systems. In MIS Quarterly, Vol. 20, No. 3, 1996; pp. 297-300.

[Lan09] Lankhorst, M. et al.: Enterprise Architecture at Work – Modelling, Communication and Analysis. (2nd edition), Springer, Dordrecht, 2009.

[LF08] Ludwig, A.; Franczyk, B.: COSMA – An Approach for Managing SLAs in Composite Services. In (Bouguettaya, A. et al. Eds.): ICSOC 2008 Proceedings, LNCS 5364, 1-5 Dec, Sydney, Australia, 2008; pp. 626-632.

[LM89] Lederer, A.; Mendelow, A.: Coordination of Information Systems Plans with Business Plans. In Journal of Management Information Systems, Vol. 6, No. 2, 1989; pp. 5-19.

[Mar82] Martin, J.: Strategic Data-Planning Methodologies. Prentice-Hall, New Jersey, 1982.

[Mar89] Martin, J.: Information Engineering. Prentice Hall (Three Volumes), New Jersey, 1982.

[PC06] de Pádua Alburquerque Oliveira, A; Cysneiros, L.M.: Defining Strategic Dependency Situations in Requirements Elicitation. In Proc. of WER2006 – Workshop em Engenharia de Requisitos, 13-14 Jul, Rio de Janeiro, Brazil, 2006; pp. 12-23.

[PS78] Pfeffer, J.; Salancik, G.R.: The external control of organizations: A resource dependence perspective. Harper & Row, New York, 1978.

[Sei09] Seifert, M.: Collaboration Formation in Virtual Organisations by applying prospective Performance Measurement. PhD thesis, University of Bremen, Bremer Schriften zur integrierten Produkt- und Prozessentwicklung, Band 63, Verlag Mainz, 2009.

[Sho93] Shostack, G.L.: How to design a service. In European Journal of Marketing, Vol. 16, No. 1, 1993; pp. 49-63.

[SP08] Scheibenberg, K.; Pansa, I.: Modelling dependencies of IT Infrastructure elements. In Proceedings of BDIM 2008 – Business-Driven IT Management, 7 Apr, Salvador, Brazil, 2008; pp. 112-113.

[SZ92] Sowa, J.F.; Zachman, J.A.: Extending and formalizing the framework for information systems architecture. In IBM Systems Journal, Vol. 31, No. 3, 1992; pp. 590-616.

[TJ01] Thoben, K.D.; Jagdev, S.: Typological issues in enterprise networks. In Production Planning & Control, Vol. 12, No. 5, 2001; pp. 421-436.

[Tho67] Thompson, J.: Organizations in action. McGraw-Hill, New York, 1967.

[Wie03] Wieringa, R.J.: Design Methods for Reactive Systems. Morgan Kaufmann, San Francisco, 2003.

[Wie08] Wieringa, R.J.: Operational Business-IT Alignment in Value Webs. In (Kascheck, R. et al. Eds.): Information Systems and e-business technologies, 2008; pp. 371-378.

[WR04] Weill, P.; Ross, J.W.: IT Governance – How Top Performers Manage IT Decision Rights for Superior Results. Harvard Business School Press, Boston, Massachusetts, 2004.

[WS09] Winkler, M.; Schill, A.: Towards dependency management in service compositions. In (Filipe, J. et al. Eds.): Proc. of the Int. Conf. on e-Business, Milan, Italy, 2009; pp. 79-84.

[Qui99] Quinn, J.B.: Strategic Outsourcing: leveraging knowledge capabilities. In MIT Sloan Management Review, Vol. 40, No. 4, 1999; pp. 9-21.

[YM93] Yu, E.; Mylopoulos, J.: An Actor Dependency Model of Organizational Work – With Application to Business Process Reengineering. In Proc. of COOCS'93; pp. 258-268.

[Zac82] Zachman, J.A.: Business Systems Planning and Business Information Control Study: A comparison. In IBM Systems Journal, Vol. 21, No. 1, 1982; pp. 31-53.

[Zac87] Zachman, J.A.: A framework for information systems architecture. In IBM Systems Journal, Vol. 26, No. 3, 1987; pp. 277-293.

[Zac08] Zachman, J.A.: The Zachman Framework: The official concise definition. 2008; URL: www.zachmaninternational.com/index.php/the-zachman-framework, Access: 22.6.2010.

[ZST10] Zarvić, N.; Seifert, M.; Thoben, K.-D.: A task-resource dependency perspective on partner selection during the formation of networked business constellations. In Int. J. Networking and Virtual Organisations, Vol. 7, No. 5, 2010; pp. 399-414.

[ZWE08]Zarvić, N.; Wieringa, R.J.; van Eck, P.: Checking the Alignment of Value-based Business Models and IT Functionality. In Proc. of the 2008 ACM Symposium on Applied Computing (SAC'08), 16-20 March 2008, Fortaleza, Ceará, Brazil, 2008; pp. 607-613.