Organizational Modularity in Enterprise Architectures: The Case of a Public Broadcasting Company

Philip Huysmans[*1], Jan Verelst[2]

Normalized Systems Institute,

Department of Management Information Systems,

University of Antwerp,

Prinsstraat 13, Antwerp, Belgium

[*1]philip.huysmans@ua.ac.be; [2]jan.verelst@ua.ac.be

Abstract- **In volatile and customer-driven markets, the ability to innovate is a key success factor. However, the innovation process is only poorly understood. Therefore, many organizations and governments have difficulties stimulating and managing innovation. Enterprise architecture frameworks allow the modeling of different layers of an organization. The ability to apply changes to these different layers should enable the implementation of innovations. However, few organizations can be considered to have completely decoupled organizational layers. Consequently, changes to one layer will impact other layers as well. In this paper, we use modularity theory to make this coupling explicit, and discuss a case study to illustrate issues and solutions related to the coupling between enterprise architecture layers.**

Keywords- Organizational Modularity; Enterprise Engineering; Enterprise Architecture

I. INTRODUCTION

Contemporary organizations are faced with rapidly changing environments. As a result, innovations need to be introduced in the organization in order to remain competitive in these markets. The introduction of innovations often impacts many different aspects of the organization. For example, Barjis and Wamba describe the impact on organizations caused by the introduction of RFID technology [1]. Besides the adaptation of the business processes through, for example, BPR, it can be expected that other organizational artifacts need to be adapted as well. Data management systems need to be able to handle "enormous" data volumes [1], hardware infrastructure and software applications need to be purchased to handle accurate tracking, and customer acceptance needs to be handled by privacy commissions and marketing departments.

Enterprise architecture projects are frequently initiated to guide the change process needed to implement such innovations [2]. Enterprise architecture frameworks help to identify the different elements to which changes need to be applied. In such frameworks, models are created from different perspectives or viewpoints. Complexity from other viewpoints is abstracted away to be able to focus on the concerns of a specific viewpoint. While this approach aids the understanding of different aspects of the organization, it can lead to unexpected results when artifacts from different viewpoints impact each other. This can be observed in the case of RFID, where changes to processes can be hindered by the data structures on which they operate. When innovative processes are designed using a BPR approach, but data systems are unable to support the required data types, implementation of the new processes will be problematic. Possibly, the redesigned processes will then need to be altered in order to be supportable. Consequently, adapting processes can be impacted by the concrete data implementation, which is not completely visible in the models which are created in a process viewpoint. While aspects of this implementation are not visible in these models, they do impact the way these models can be used, and therefore need to be considered.

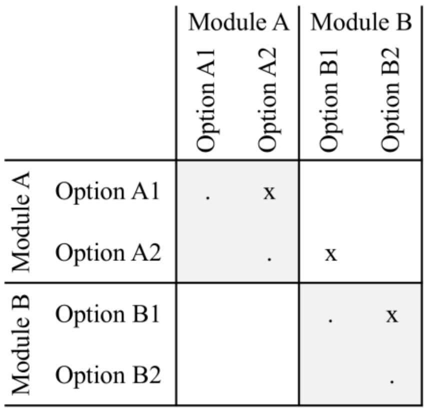
In this paper, we argue that modular dependencies can be used to explicitly identify such impacts. This approach can be applied complementary to existing enterprise architecture modeling. We introduce relevant literature for this approach concerning modularity and enterprise architecture frameworks in Section II. We then present a case study to illustrate the application of modular dependencies in organizations which are faced with innovations which require changes in several viewpoints. We therefore present the methodological approach of this case study in Section III, and introduce the organizational context of the case study in Section IV. In the case study, we focus on an enterprise architecture project in a PBC. We analyse how coupling in the lower enterprise architecture levels dictate the design of artifacts on the business level at the organization in Section V. We therefore demonstrate how the structure of lower enterprise architecture levels impacts the agility of the business level. We analyze both the situation before the enterprise architecture project, and the solution proposed in the project itself. We discuss how our observations can impact the practice of enterprise engineering in Section VI. Finally, we offer the conclusion of our work in Section VII.

II. RELATED WORK

In this section, we introduce a necessary background of modularity and enterprise architecture research literature.

A. Modularity

Organizational modularity recently receives much attention in both research and practice. Campagnolo and Camuffo provide a literature overview of 125 management studies which use the modularity concept [3]. They define modularity as "an attribute of a complex system that advocates designing structures based on minimizing interdependence between modules and maximizing interdependence within them" [3]. They argue that organizational artifacts such as products, production systems, and organizational structures can be regarded as modular structures. A characteristic of a perfect modular structure is that it allows parallel evolution of different modules [4]. However, dependencies between the modules of such a structure limit the autonomy of the individual modules. Baldwin and Clark state that "because of these dependencies, there will be consequences and ramifications of any choice" made during the design of the artifact [4]. A design choice for a given parameter can limit or affect the possible design choices concerning other parameters. In traditional modularity approaches (e.g., product modularity), dependencies between and within modules are visualized by Design Structure Matrices (DSM). In a DSM, an artefact is described by a set of design parameters. The matrix is then filled by checking for each parameter by which other parameters it is affected and which parameters are affected by it. The result is a map of dependencies that represent the detailed structure of the artifact.



		Module A		Module B	
		Option A1	Option A2	Option B1	Option B2
Module A	Option A1	.	x		
	Option A2		.	x	
Module B	Option B1			.	x
	Option B2				.

Fig. 1 An example Design Structure Matrix.

An example design structure matrix is shown in Figure 1. Dependencies are represented by an "x". The intersection of identical design options is marked with a ".". Consider the design dependency which is represented by the "x" in the intersection of the column of design option A2 and the row of design option A1. This signifies that design option A2 influences design option A1: the design decision for design option A1 will be dependent on the decision taken for design option A2. This dependency does not break the modular structure of the artifact, since design options A1 and A2 both belong to the same module. Now consider the dependency of design option B1 on design option A2. Since these design options belong to different modules, it can be concluded that these modules are directly dependent on each other. Therefore, this dependency does violate the modular structure. Indirect or chained dependencies can occur as well. While design option B2 does not seem to affect any design options of module A, it does affect design option B1. As we discussed before, design option B1 does affect design option A1. Therefore, a so-called chained dependency exists between design options B2 and A2.

Originally, DSMs were used to model modular products. In later publications, a DSM was also applied to model organizational departments [5]. This indicates that similar tools can be used for analyzing modularity on various levels. In this paper, we will focus on modularity on the organizational and software levels. Organizational modularity focuses on other artifacts than product modularity within the same organization. Research on this level has been performed by, for example, Galunic and Eisenhardt [6], who consider organizational divisions as modular organizational building blocks. This kind of modularity therefore enables the flexibility required on a business level. On the software level, modularity is used to achieve a flexible structure as well. For example, Parnas argued that a modular decomposition in software systems should be made to isolate the impact of changes [7]. When the impact of a change remains within a module, changes can be applied to individual modules without requiring changes in the rest of the system. While modularity is applied to both the organizational and IT level by various researchers, most research projects focus on a single level. However, interactions between a modular approach on the organizational and the software level remain an important issue. Given the high dependence on IT systems, it is important that a change on the organizational level (e.g., changing organizational divisions) can be handled by the supporting IT systems as well. Otherwise, the inability to change the IT systems will restrict the ability to change organizational modules. Indeed, modularity needs to be considered as a relative attribute of complex systems (such as organizations), meaning that within a single artifact, different levels of modularity can exist [8].

B. Enterprise Architecture

Enterprise Architecture (EA) frameworks provide insight to the structure of organizational goals, divisions, and supporting IT systems. By specifying separated viewpoints on organizational artifacts, an overview is provided of specialized models created by different stakeholders [9]. Most enterprise architecture frameworks use a top-down perspective. They start by defining business goals and a high-level artifacts to realize these goals (e.g., an organizational structure). Based on these artifacts, lower-level artifacts are defined which offer services to support the business level. The following architectural levels are usually identified [10]:

- Strategy layer
- Organization layer
- Integration layer
- Information layer
- Application layer
- Infrastructure layer

For example, TOGAF suggests to use an iterative process consisting of eight phases to develop an enterprise architecture. Based on the business goals which are defined in the first phase, different architectures need to be developed in the following order: business architecture (second phase), information systems architecture (third phase), and technology architecture (fourth phase). These architectures correspond to the defined layers. The business architecture is defined on the business layer. The information systems architecture consist of both the functional and information layer. The technology architecture is addressed on the infrastructure layer. This approach assumes that supporting services can be created straightforwardly based on business requirements. In a literature review on enterprise architectures, Lucke et al. identify several issues related to this approach [11]. First, complexity is referred to as an underestimated issue. Not only the complexity of the models themselves, but also the dependencies between the different layers remain problematic. Second, rapidly changing conditions imply that a top-down specification of an enterprise-wide architecture can become out-dated before it is even implemented. Third, a top-down specification of the architectural layers results in issues regarding scoping of architectural descriptions. Rather than being straightforward, the identification of organizational and technical services to support business requirements is often considered to be problematic. Therefore, a complementary approach is required to represent the impact of lower-level layers on higher-level layers.

III. METHODOLOGY

In the case study, we apply the related work presented in Section II. The unit of analysis in this case study is the enterprise architecture project conducted at a Public Broadcasting Company (PBC). This scope is suitable for a case study, since it allows to study the enterprise architecture project in its concrete business environment [12, 13]. This is important, because the specific influence of the business context may be relevant to the observations and conclusions in this case study. Therefore, we selected a case study approach since it is well suited for researching a phenomenon where the boundaries between the phenomenon and context are not clearly defined [12]. The PBC is faced with important changes in its target market. We focus on the division which is responsible for broadcasting news journals. Because of changing technology, the way in which customers demand access to news has changed. In order to react to these changes, the organization needs to be adapted in several aspects. Therefore, an enterprise architecture project is initiated. The architectural innovation currently focuses on the news broadcasting division, but once the new architecture is operational and successful here, it will be expanded to all production areas of the PBC.

Since we perform an explorative case study, we adhere to the guidelines as formulated by Yin [12]. The primary mode of data collection consisted of face-to-face interviews. We have used the key informant method to identify three informants who were highly knowledgeable about and involved in the enterprise architecture projects. Our first informant is a member of the former R&D department of the PBC. While the R&D department is currently an independent organization, it continues to cooperate closely with the PBC. This cooperation is important, since the R&D organization can explore new technical solutions (1) without being restricted by the current implementation of technologies in the PBC, and (2) with the ability to change substantial components of the architecture without needing to ensure the continuation of broadcasting activities. The other two informants are employed directly by the PBC, and are both knowledgeable of the IT projects and strategy development. The second informant is responsible for aligning the business requirements with the architectural IT projects. Therefore, he can be positioned at the business side of the organization. The third informant is a member of the IT department of the PBC, and is the chief architect for the observed platform. The interviews have been digitally recorded for future reference, and have been transcribed in order to facilitate coding and cross-case analysis. Before the interviews, various documentation sources have been consulted to achieve an initial understanding of the organization. These consist of press releases, product offerings for media providers, and papers published by the R&D organization. During the interviews, documentation from a variety of sources has been collected in order to be able to validate the interpretation of the interviews. This documentation consists of internal presentations, specifications of the architecture, and business process models.

IV. CASE DESCRIPTION

Traditionally, this news division of the PBC offers radio and television transmissions, which follow a clearly defined schedule. The radio and television business units import news items from different sources, such as feeds from external agencies, or items made by reporters. Items can be imported using physical tapes, digital files or through satellite transfers. Therefore, incoming items can be characterized in three different dimensions: the nature of the item (i.e., audio or video), the source of the item (i.e., external agencies or internal reporters), and the medium (i.e., tape, digital or satellite). For a large part of the combinations of these three dimensions, different applications are in use. Once the items are imported, they are edited and transmitted in the form of a news journal. On the business level, the television and radio departments are the two major constituents in the organizational structure. They are relatively independent from each other, with their own reporters, editors, and support systems. In terms of IT support, they rely on packages from different vendors, who specialize in the audio or video market. In this case study description, we will position the discussed artifacts on the enterprise architecture layers identified by [10] (i.e., strategy, organizational, integration, information, application, and infrastructure layers). For the PBC, the radio and television business units are therefore situated on the organizational layer of their enterprise architecture.

Our respondents claim that “the fast-paced evolution in computer technology, and more specifically, in network technology, has resulted in changes in this market”. More specifically, they stress the increased competition from other networks, the new customer preferences for media consumption, and the increasing importance of internet force many organizations to change. Especially the increased network capacity, which allows end-users to view news items over the internet on a variety of devices, is of relevance to the news division. Customers now demand personalized and real-time access to transmissions, so that even the concept of a news journal needs to be reconsidered. Consequently, the organization of news items in the journals needs to be approached differently. Moreover, the contents of the news items themselves change. On top of the item itself, customers expect the usage of additional media types (i.e., text, audio and video), and references to related news-items. Our respondents believe that “continuously increasing diversification in media production is inevitable”. In order to support such diversification, it is important that news items can be reused outside the context they have been originally created for. Currently, the way of working with news items is dictated by the structure of the software that is used to import the item. Therefore, changes in response to the business environment are not easy to make.

In response to the changing market environment, the PBC decided to create a dedicated business unit for providing an online channel, next to the existing radio and television units. This new business unit could reuse content from both the radio and television units, and create dedicated online news items as well. Adding the online channel was considered to be a necessary strategic move in order to serve a changing market. In order to provide an adequate response for the changing customer demands, the key characteristics of this new business unit should be: a faster processing of incoming news items, the reuse of existing news items, and a focus on a cross-media approach. According to our respondents, these goals are formulated on the organizational layer of the enterprise architecture.

For the organization, it was clear that the support of the new business unit required a “digital way of working”. However, it was also clear that the current architecture of the PBC was not able to support the addition of an extra news channel. In the current situation, the TV and radio business units operated independently from each other. The produced news items in the TV department were mostly archived on physical tapes, while the news items created in the radio business unit are archived in a self-contained software package. In both situations, the medium on which the news items were stored had important consequences on the way in which they could be reused. An item which is archived on a tape cannot be accessed directly, because a tape does not allow random access: it needs to be accessed sequentially. Consequently, the time and effort required to access a certain news item were dependent on the presence of the other news items on that tape. Moreover, the need for the simultaneous use of news items which are on the same tape results in issues. Another aspect was the inability to include meta-data with the actual content of the news item (called the essence). Information concerning the news items on a tape was noted on a piece of paper attached to the tape. A separate application was in use, which contained all the meta-data of the news items, but this application can only refer to the location of essence items. Consequently, no integration of essence and meta-data was available. The self-contained software system for archiving radio items did provide an integration for essence and meta-data, but is very complex and is not integrated with other systems. We will elaborate on the concrete obstacles and their implications in Section V-A.

Based on the limitations of the current architectural approach, the PBC decided to initiate a project to radically change the enterprise architecture layers supporting the organizational layer. Because they required an architecture which allowed them to reorganize their business operation easily and decouple the concepts used on the business level (i.e., news items) from their concrete implementations, they became interested in a Service-Oriented Architecture (SOA) approach. We will describe the aspects from SOA which have been used in this project in more detail in Section V-B. In order to be able to evolve towards a more decoupled architecture, the PBC decided to develop a centralized, file-based system to organize the importing, sharing and archiving of all news items. Therefore, this system mainly focuses on the storage concern. The goal of this system is to be able to transport news items between different work centres. A work centre is an autonomous environment used for performing a specific craft step in the production process. Examples of such work centres are the actual broadcasting of the finished news journals (i.e., play-out), modifying and editing audio, modifying and editing video, creating playlists of news items for news journals, creating narratives for news readers which link to news items, transcoding, etc. An important principle of the PBC is to select an optimal software package for each work centre. Because of the large variety of functionality in the work centres, these software packages are often created by different vendors. Therefore, the different software packages do not integrate well.

Consequently, the integration will need to be enabled by the central platform itself. Our respondents indicated that "this decision allowed for a clear selection of individual software packages, but resulted in a large complexity for the integration architecture". Every single software package that needs to be connected with the platform is expected to result in a substantial amount of integration complexity.

In order to transport news items between different work centres, they need to be available digitally. Moreover, a combination of the essence data of the news items, as well as its meta-data needs to be transferred. Currently, the integration which exists between different software packages only considers the essence, not the meta-data. As a guiding principle, the incoming news items all need to become file-based. As discussed, news items could be imported from tapes, file-based, or via satellite streams. If a news item is not imported as a file, it needs to be converted to a file as early in the process as possible. The biggest issue in this conversion is not the technical conversion of the essence data itself, but the correct conversion of the meta-data.

V. CASE RESULTS

In this section, we elaborate on the case study results. We first demonstrate how the issue addressed in the enterprise architecture project can be understood more clearly from a perspective using modularity theory. Then, we analyze the proposed solution using this perspective.

A. Understanding the Issue

Adding this additional business unit posed serious problems due to the supporting structure of the PBC on the lower enterprise architecture levels. In the previous section, we described how it would be difficult to use the existing archives to reuse news items. However, the PBC business users required even faster reuse of the news items. The news items are archived only after they have been broadcasted. Instead, the business unit for the online channel needs to be able to reuse these items as soon as they are imported. Only then, they could achieve real-time news coverage. Accessing imported radio and television items proved to be complex, since they were stored in specialized applications. For almost every combination of import channel (i.e., tape, digital or satellite), media type (i.e., audio or video) and source (i.e., reporters or news agencies), different applications were in use. The applications to import the items contained functionality to perform basic editing as well. Therefore, the item usually resides in the application it was imported in. If the item needs to be retrieved to be reused, one has to be able to use that specific application. Moreover, additional elements hinder the reuse of news items. In order to analyze these elements, we need to look at the implementation of the current business units, in various dimensions. In the previous section, we argued how the implementation medium has large consequences on the reuse of news items.

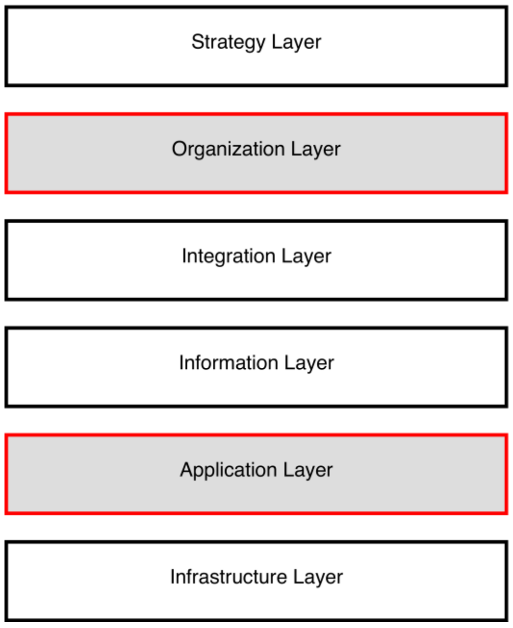


Fig. 2 Positioning the PBC case study in enterprise architecture layers

The wide diversity of used applications is a direct result of the principles used to develop the application portfolio, which is located on the application layer in the enterprise architecture. As a general principle, the organization always selects best-in-class software solutions for specialized media editing. This principle ensured the most efficient editing process. However, this results in an application portfolio which is not well integrated. Employees with specialized competences are required to operate these software packages. Therefore, in order to reuse audio and video fragments for the online channel, employees with these competences needed to be made available for the online business unit. When the TV and radio business units operated independently, this was not an issue: employees with the skills and competences are required to be available to import the incoming items. However, when additional employees needed to be trained or hired for the online business unit, this resulted in a duplication of skill sets, which was not efficient. The inclusion of employees with the competence to use the specialized software packages was not foreseen when the new business unit was defined. However, the coupling between software packages and employee competence on the information layer necessitates this inclusion. Put it differently, the required competences defined on the organizational layer need to be adapted to account for a dependency with the application level. Therefore, coupling with the application layer is solved by adapting the artifacts which are defined on the organizational layer. Consequently, the ability to take strategic decisions to serve an emerging market is impacted by decisions made on the application layer. While the principle to select best-in-class applications may be justified for this sector and the performance of the organization, the lack of integration which it causes and the restrictions it places on business flexibility need to be understood as well.

The specification of the principle “Select the best-in-class software package for each work centre” is consistent with the approach used in some enterprise architecture frameworks. For example, as an example application architecture principle, TOGAF suggests the use of the principle “Ease of use” [9, Section 23.6.3]. As discussed, TOGAF uses a top-down approach, and uses general organizational architecture as input for the application architecture [9, Section 11.3.3]. TOGAF mentions the possibility that impacts with other architectural aspects may occur. However, this is only mentioned: no specific approach or guidance is provided (e.g., “Resolve Impacts Across the Architecture Landscape” in Section 11.4.6 is the most specific guideline). In order to position the observed issues better in relation to the surveyed enterprise architecture literature, we included Figure 2. In this figure, we highlighted the affected layers, i.e., the application layer and organizational layer, to indicate the origin and observed effect of the issues. It should be noticed that before the enterprise architecture project, none of the other layers were explicitly addressed.

		Business Unit				Work Center			
		Organizational chart	Charter	Media reuse rate	News item granularity	Competence	Platform	Storage medium	Essence data collections
Business Unit	Organization chart	.	x			x			
	Charter	x	.						
	Media reuse rate		x	.	x			x	
	News item granularity		x	x	.			x	x
Work Center	Competence					.	x		
	Platform						.	x	
	Storage medium						x	.	
	Essence data collections							x	.

Fig. 3 The DSM for PBC.

Our respondents considered “Select the best-in-class software package for each work centre” as an enterprise architecture principle on the application layer based on the use of the TOGAF framework. However, in the current implementation of the PBC organization, this principle impacts aspects of the organizational EA layer as well. Currently, no approach is available to ensure that the impact of principles for a certain design domain only restricts that design domain. Indeed, this issue has been reported in various empirical enterprise architecture studies. Various authors have reported the occurrence of local optimizations [14, 15, 16], which do not contribute to a general optimization for the organization, or which limit or restrict the design of related organizational domains. In this case study, this principle indeed resulted in a correct decomposition on the application layer: for every import source and every media type, a separate software package is required. However, because the elements from this application layer are coupled with the organizational layer, the same composition is forced on this level. As discussed, this severely limits the flexibility on the organizational layer.

Consequently, the notion of modular dependencies seems to gain an understanding of the issue presented here. The required competence of employees is a modular dependency which can be observed between various layers of the enterprise architecture. According to modularity theory, it is clear that such dependencies should not occur. The GSDP shows that each layer needs a functional and constructive perspective. When an application is added to import videos, an employee needs to be added in each business unit which needs to be able to use video news items. The functionality required by the organization (i.e., consulting news items) does not change. However, because the design of the application layer is not appropriately encapsulated, design aspects of this layer impact the organization as well. While the number of impacts may not be very high, the size of the impact is large: an employee with the required skills needs to be added to the business unit. Using modularity terminology, the dependencies between aspects of modules on the application layer (i.e., applications) and modules on the organizational layer (i.e., business units), can be represented using a design structure matrix (DSM), as introduced in Section II. The DSM for this single dependency is presented in Figure 3. It shows how a design parameter from the applications, which is part of the application layer, affect the organization chart design parameter of the business unit, which is part of the organizational layer.

Moreover, modular dependencies can be identified as well within a certain enterprise architecture layer. Our respondents indicated that the concerns of security, scheduling, meta-data, logging and transcoding can be considered to be change drivers on the application level. Because of the selection of different applications, these concerns are implemented differently in the various applications as well. Each time an application is added, integrations with all these concerns for every available application need to be developed. It is clear that the impact of such an additional application grows as the number of applications grow. These dependencies are represented in Figure 4.

Using our proposed theoretical perspective, a structured analysis can be made of the issues which occurred at the PBC related to the integration of different enterprise architecture layers. In Figure 4, the concerns from the applications which need to be handled on the application layer are represented. These concerns can be positioned in the application layer. However, Figure 3 shows that dependencies between artifacts from different layers do occur as well. While these dependencies are not considered in enterprise architecture frameworks, a design structure matrix allows us to make these dependencies explicit.

		Application A					Application B				
		security	scheduling	meta-data	logging	transcoding	security	scheduling	meta-data	logging	transcoding
Application A	security	.	x	x	x		x		x		
	scheduling	x	.			x		x			x
	meta-data			.	x				x	x	
	logging	x			.		x			x	
	transcoding			x	x	.			x		x
Application B	security	x		x			.	x	x	x	
	scheduling		x			x	x	.			x
	meta-data			x	x				.	x	
	logging	x			x		x			.	
	transcoding			x		x			x	x	.

Fig. 4 The DSM for the applications of PBC

B. *Understanding the Solution*

Since the PBC was unable to motivate the costs of specialized employees for the new channel, a structural solution for the implementation was proposed. The dependency on the required application competences forces the organization to deal with concerns from the application layer on the organizational layer, i.e., the different handling of tape, digital and satellite items. This dependency was removed by developing an abstraction layer on top of the application layer, which provided only the functionality needed on the organizational layer. This abstraction layer was defined based on concepts known in the organizational layer: the basic entity on this abstraction layer was a news item.

In order to be able to provide a uniform access, a central media hub was developed. Before the development of this central media hub, all applications which needed to interact with each other needed a direct interface between them. As discussed in the previous section, various concerns could be identified which made this integration cumbersome: for every application, these concerns could be implemented differently, and different integration logic needed to be provided. It could be argued that this solution correlates with basic principles of engineering: instead of applying point-to-point integration, a bus pattern is applied. Moreover, a more asynchronous way of dealing with news items is introduced, since the news items are available immediately after being imported. In contrast, news items used to be available only after the entire process of importing, editing and broadcasting was completed. This required to be able to handle the essence data and the meta-data as one integrated object. When direct integration between the applications was implemented, only the essence data was transferred. On the application level, the essence data is of a completely different nature than the meta-data, and need to be dealt with separately. However, for business users, the combination of essence data and meta-data is considered to be one coherent whole.

Therefore, a *transport/control layer* is developed to provide a generic transport mechanism. This layer handles the physical transport of both essence data and meta-data between the systems of different work centres. In this layer, it is explicitly assumed that meta-data needs to be gradually enriched throughout the different steps of the production lifecycle of the media content. Therefore, this layer is designed with changes in meta-data in mind. Therefore, so-called correlations between versions of the meta-data are maintained. If the meta-data is changed in the source system, the meta-data in the target system needs to be updated as well. When considering the essence data, all concerns mentioned in the previous section need to be accounted for. For example, different encoding types could be used on the source and target systems. Consequently, the transport/control layer needs to offer transcoding services to ensure the delivery in the correct encoding type. Logging facilities are provided as well to monitor the life cycle of the news items. Therefore, the allowed activities of users on news items can be logged, and failures can be reported. These failures can now be reported centrally to a support department, which can react accordingly. Based on the type of problem, the failure can be fixed on the level of the news item, or on the level of the application. Because of the asynchronous way of reporting errors, a single support centre can be used for all departments.

While we focus on technical aspects, our aim is to describe how on the transport/control layer, an adequate key element is defined, and its related concerns are completely handled on that specific level. This is necessary to ensure that these concerns do not propagate to the higher-level layers, impacting organizational aspects as described above. The granularity of the key element on this level was determined to be a single news item, incorporating both essence data and meta-data. Many applications use other levels of granularity, such as collections of news items. These collections can contain different news items with similar characteristics (e.g., radio-items transmitted on a certain day), but can also refer to aggregated objects: different video-items, which are combined in a journal with transition effects and a single audio item. However, supporting such collections on this level resulted in several issues, such as the lack of traceability of correlations between meta-data. Consequently, dedicated objects on the transport/control layer are developed which offer services in terms of the news items, hiding the actual implementation of the different concerns. Examples of such functionality are: get/set meta-data, get/set essence data, and createNewsItem. Technically, this functionality can be considered as an interface which hides the specificity of SOAP calls or the XML/FTP combination used by individual applications.

A news item entity abstracts from the method which was used to import the source material. Therefore, it served as a kind of interface to the specialized software applications. For every application, audio and video fragments can be extracted, together with the required meta-data, in a uniform way. The specific functionality of the editing programs is not available through this interface, since it is not required on the organizational level. The specific functionality of the applications and their related complexity should not impact the organizational level. In order to use the editing functionalities, specialized competences remain necessary. However, the need for employees with these competences is now not imposed on the organizational layer, but on the application layer. Consequently, this modular dependency can be considered to be resolved.

On top of the transport/control layer, a so-called business layer was constructed which allowed business users to interact with the elements on the transport/control layer. Based on events, such as the import of a certain type of news item or the request for a certain item, the business layer use the functionality of the transport/control layer. On this business layer, a pure focus on how end-users interact with news items can be developed. They can request certain news items, edit them, etc., without being restricted by the implementation of concerns of the specific applications. It is clear that on the functional side of the business layer, no knowledge of the actual systems is present. Only the interaction between end-users is considered here. In the future, more elaborate processes can be developed on this layer. The addition of the transport/control and the business layer can be considered as a necessity to fill the gap between the organizational and application layers. This gap was discussed above, and graphically represented in Figure 2.

However, it is clear that it is very difficult to separate all concerns within the layers. While many dependencies have been resolved, many dependencies remain. As an example of a remaining dependency, we discuss the usage of the meta-data format. We already illustrated in the design structure matrix why the selection of a single internal meta-data format was crucial for decoupling different applications. Otherwise, conversions between meta-data formats used in specific applications needed to be developed. The PBC decided to use an industry standard for this internal meta-data format (i.e., P/META). Currently, the major advantage of using the P/META standard is the selection of a single internal format. The decoupling of the business layer, which allows the PBC to develop business processes based on the interactions of end-users with news items, should allow easier integration with external partners as well. The PBC assumed that following an industry standard would improve flexibility. However, the explicit coupling of the internal meta-data format to the industry standard already resulted in issues when integrating with external partners. For example, changes to the current version of the standard are very hard to implement, since the integrating platform needs to change every integration implementation with the applications. Therefore, if an external partner uses a different version of the P/META standard, integration becomes as complicated as integrating with a completely different meta-data format. P/META is a standard which has been developed by a consortium of organization. Because of the various interests of these organizations, the description of the meta-data fields is rather generic. Therefore, many organizations develop customizations for the standard. Currently, the PBC uses around thirty meta-data fields. However, a competing standard is currently being developed, which is promoted by a large industry vendor from international markets. Consequently, PBC expects that all external partners could use that new standard. However, such a change would require many changes in the integration platform. This example shows that the currently defined elements are still very coarse-grained. In modularity theory, the meta-data format would be considered to be an external change driver, and should therefore be contained.

VI. CASE DISCUSSION

This case shows that decisions taken on the application layer of enterprise architectures can make changes of the organizational layer more complex. By adequately encapsulating the complexity from the application layer, dependencies between the different layers can be removed, and necessary services can still be offered. As a result, decisions on the organizational layer only need to deal with the inherent complexity imposed by the business environment, instead of dealing with complexity originating from the supporting layers as well. In this case, the number of modules is low and the dependency which limits the organizational layer is clear. However, this case study provides a clear illustration of the issue we address in this paper. The impact of dependencies originating from supporting layers was larger than assumed in enterprise architecture frameworks. These dependencies were the very reason for initiating the enterprise architecture project.

As discussed, encapsulating concerns in separate modules is hard. Based on the modularity analysis performed in this paper, the following issues can be identified.

First, the identification of remaining modular dependencies has shown that coupling between different layers may still exist. In the previous section, we focused on the meta-data syntax which was not separated from other concerns. However, several other combinatorial effects have been documented during the case study. For example, the component responsible for importing a new news item on the application layer follows a certain workflow. Most of the mentioned concerns are handled separately in this workflow. These workflows contain two elements which are general containers for handling concerns. At the beginning of the workflow, an “initTransfer” process step is provided. At the end of the workflow, a “termTransfer” process step is provided. In these process steps, concerns such as security are currently handled. However, as modularity theory shows, the bundling of these concerns introduces dependencies between modules: when security is handled in multiple termTransfer process steps, all these steps are impacted when the security handling changes.

Second, other indications can be observed which suggest that the correct reusable building blocks or modules are not yet described. For example, we analyzed the kind of changes which need to be applied when the current solution should be reused in areas other than news production. In the beginning of this case, we discussed how the current architecture needs to support other production areas as well. We already demonstrated how the format of meta-data is currently not encapsulated. Our respondents indicated that different fields of meta-data are required for, for example, music items or television shows. However, the current architecture did not provide support for two different versions of meta-data to be used.

Third, the introduction of the integrating hub introduces an additional risk of failures in the broadcasting process. Therefore, the PBC decided to preserve several applications and their integration components as an emergency backup. Consequently, redundancy in the functionality of some applications may occur. Moreover, it is unclear how changes to these applications will be handled: while it is crucial for the PBC to have an emergency backup system, our respondents claim that they “are aware of the large complexity when changing directly integrated applications.” For example, the video-storage system is still operational, and can be used when the central platform is not available. This system can interact directly with the broadcast system. However, items which are used during this period are not included in the central system, and need to be synchronized later.

These issues demonstrate that it is not easy to develop modular enterprise architectures. In this paper, we proposed the use of design structure matrices to analyze modular dependencies in enterprise architecture projects. Moreover, the following observations can be made.

First, a modular architecture seems to be able to increase flexibility in real-life projects. Notwithstanding this observation, current enterprise architecture frameworks do not seem to provide sufficient guidance to develop modular enterprise architectures. In the case study, agility on the business level has been improved because of the underlying architecture. Our respondents indicated that an increase of efficiency has been achieved because of the decoupled architecture. This applied especially to the online department. Moreover, the TV and radio departments are now able to create programs which were impossible to produce before. For example, different news items which have been imported during the day are now bundled and discussed with a guest commentator. Therefore, the end-users now can react more quickly to new preferences or ideas. This is attributed to the flexibility of the architecture itself, rather than to the actual software systems. While the PBC was originally interested in SOA, our respondents claim that “only little guidance on how the actual problems, such as the correct identification of the different layers and key concepts, need to be handled is available in the frameworks”. This case study shows that the top-down development of architectural layers, as proposed in most enterprise architecture frameworks, is difficult to achieve when an organization has already been implemented and operational. During the design of higher-level layers, no attention is usually given to the restrictions imposed by the lower layers. For example, the TOGAF ADM prescribes to develop the business architecture first (i.e., in phase B), in order to describe how business processes in a to-be version of the organization should operate. Only when this phase is completed, the IT architecture to support these business processes should be developed (i.e., in phase C). Therefore, restrictions imposed by the IT architecture only become apparent later in the architecture process. In the Zachman framework, the notion of constraints is introduced, and it is acknowledged that these constraints are additive in a bottom-up way: higher-level layers need to consider the aggregated constraints of lower-level layers. However, [17] argue that “the designers who are responsible for the two rows must initiate a dialog to determine what must be changed and to ensure that no gap in expectations exists between the different perspectives”. Therefore, no structural solution is provided. In this case study, it is clear that the awareness of the origin of the constraints is not enough to adequately address the constraints. This motivates the scope of our research project to include both purely organizational, non-IT issues, as well as some IT-issues, particularly those that are related to the business-to-IT transformation. This case study shows that the design on the organizational layer of an enterprise architecture may vary from being naively optimistic about implementation to even being impossible to implement.

Second, while frameworks do not seem to guide the actual architecting process, many generic engineering insights have been applied in this case. For example, it was discussed how the import processes put restrictions on the usage of news items: news items were archived after they had been broadcasted, and could not be reused before this archival. Currently, all news departments have access to all news items from the moment they are imported in PBC. While the former way of working resembles the synchronized way of processing, the current situation has more in common with the asynchronous way of working. We discussed how other organizational units such as the support organization could be optimized as well because of this asynchronous way of working. When a failure occurs with a news item, the support organization can check its history in the logger, and identify issues more accurately. Such analysis was not feasible when applications were coupled directly to each other, and a synchronous way of working was employed. The use of asynchronous processing has been considered to be beneficial in other engineering fields, such as software engineering [18]. Other examples of the use of engineering insights are the usage of a bus pattern, instead of using point-to-point connections for the work centres. These examples illustrate the relevance of general engineering insights in other domains. Therefore, the integration of engineering knowledge in enterprise architecture frameworks should be encouraged.

VII. CONCLUSIONS

In this paper, we focused on the lack of agility in a real-life organization. We identified the coupling between layers in the enterprise architecture as an important cause of this issue. Therefore, we proposed the use of modularity theory. We analyzed the situation before the enterprise architecture project, as well as the project itself, using modularity tools such as the design structure matrices. We showed that a modularity analysis can indeed identify relevant issues in enterprise architecture projects. Moreover, we discussed several remaining issues. First, we showed how difficult it is to remove all modular dependencies. Second, we argued that truly reusable building blocks or modules are hard to design. Third, we noted that the introduction of additional layers can actually complicate the architecture, or introduce new risks and modular dependencies.

While enterprise architects expect guidance to resolve these issues from frameworks, only a limited amount of such guidance seems to be available. Nevertheless, we observed in the case study that well-known engineering insights from other fields can be applied to enterprise architecture as well. In future work, we will therefore make use of these engineering insights in enterprise architecture projects more explicitly.

REFERENCES

[1] J. Barjis and S. F. Wamba, “Organizational and business impacts of rfid technology,” Business Process Management Journal, vol. 16, no. 6, pp. 897–903, 2010.

[2] J. Schekkerman, “Trends in enterprise architecture: How are organizations progressing?” Institute For Enterprise Architecture Developments, Tech. Rep., 2005.

[3] D. Campagnolo and A. Camuffo, “The concept of modularity in management studies: A literature review,” International Journal of Management Reviews, vol. 12, no. 3, pp. 259–283, September 2010.

[4] C. Y. Baldwin and K. B. Clark, Design Rules, Volume 1: The Power of Modularity, ser. MIT Press Books. The MIT Press, January 2000.

[5] C. Y. Baldwin and K. B. Clark, “The value, costs and organizational consequences of modularity,” May 2003, working Paper. [Online]. Available: http://www.people.hbs.edu/cbaldwin/DR1/DR1Overview.pdf.

[6] D. C. Galunic and K. M. Eisenhardt, “Architectural innovation and modular corporate forms,” The Academy of Management Journal, vol. 44, no. 6, pp. 1229–1249, 2001.

[7] D. L. Parnas, “On the criteria to be used in decomposing systems into modules,” Communications of the ACM, vol. 15, no. 12, pp. 1053–1058, 1972.

[8] H. A. Simon, “The architecture of complexity,” Proceedings of the American Philosophical Society, vol. 106, no. 6, pp. 467–482, 1962.

[9] The Open Group, “The open group architecture framework (togaf) version 9,” 2009, http://www.opengroup.org/togaf/.

[10] M. Sch ӧenherr, “Towards a common terminology in the discipline of enterprise architecture.” in ICSOC Workshops’08, 2008, pp. 400–413.

[11] C. Lucke, S. Krell, and U. Lechner, “Critical issues in enterprise architecting - a literature review,” in AMCIS 2010 Proceedings, 2010.

[12] R. K. Yin, Case Study Research: Design and Methods, 3rd ed. Newbury Park, California: Sage Publications, 2003.

[13] I. Benbasat, D. K. Goldstein, and M. Mead, “The case research strategy in studies of information systems,” MIS Quarterly, vol. 11, no. 3, pp. 368–386, 1987.

[14] D. Dreyfus, “Information system architecture: Toward a distributed cognition perspective,” in ICIS 2007 Proceedings, 2007, paper 131.

[15] J. Espinosa and W. F. Boh, “Coordination and governance in geographically distributed enterprise architecting: An empirical research design,” in 42nd Hawaii International Conference on System Sciences, 2009. HICSS ’09, pp. 1 –10.

[16] V. Seppanen, J. Heikkil, and K. Liimatainen, “Key issues in ea implementation: Case study of two finish government agencies,” in Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing, 2009, pp. 114–120.

[17] J. F. Sowa and J. A. Zachman, “Extending and formalizing the framework for information systems architecture,” IBM Syst. J., vol. 31, no. 3, pp. 590–616, 1992.

[18] H. Mannaert and J. Verelst, Normalized Systems—Re-creating Information Technology Based on Laws for Software Evolvability. Kermt, Belgium: Koppa, 2009.