# DESIGNING FOR INNOVATION
## *Using Enterprise Ontology Theory to Improve Business-IT Alignment*

Philip Huysmans, Kris Ven and Jan Verelst
*Department of Management Information Systems, University of Antwerp, Prinsstraat 13, B-2000 Antwerp, Belgium*

Keywords: Enterprise architecture, Enterprise ontology, Innovation, Information technology, Alignment.

Abstract: In today's economy, innovation plays an increasingly important role in the strategy of organizations. Managers therefore need to understand and be able to manage the innovation process. The recent research efforts in the enterprise architecture domain are very relevant in this regard. Most of these frameworks acknowledge the importance of aligning the information technology (IT) infrastructure with the enterprise architecture. In this paper, we focus on a case of an organization that was able to realize substantial business innovation by aligning its IT architecture to its enterprise architecture. Notwithstanding the successful outcome of this enterprise architecture project, the approach taken by the organization strongly relied on the heuristic knowledge of employees, thereby limiting the repeatability and reproducibility of their approach. In addition, it remains unclear whether the modeling technique that was used will be able to provide the required level of evolvability in the future. It therefore seems useful to apply a systematic method to be able to recreate these results in other organizations. We therefore take a design science approach by repeating the enterprise architecture project using the Enterprise Ontology theory. Our results show that the model created by following the Enterprise Ontology theory was very similar to the model created by the organization, which is a desirable result. The main advantage of Enterprise Ontology is that it provides a more repeatable and reproducible result and that the resulting models are more evolvable.

# 1 INTRODUCTION

In today's economy, innovation plays an increasingly important role in the strategy of organizations. Since organizations nowadays have to compete on a global level, it is important that organizations are able to generate and exploit innovations at a steady pace to seek sustainability of their business (Van de Ven and Angle, 2000). There is consensus in literature that information technology (IT) is an important enabler for innovation (Brynjolfsson and Saunders, 2010). Given the importance of innovation to organizations, it is important that managers understand and are able to effectively manage the innovation process. It has indeed been noted that "*[a]t a time when so much attention is given to innovation and entrepreneurship, it is rather pathetic that a deep understanding of the process is lacking. It is no wonder that firms and governments have difficulty trying to stimulate (and manage) innovation when its fundamental processes are so poorly understood.*" (Teece, 1987, p. 3). Although substantial progress has been made in this field, it remains remarkable that almost 25 years later, much innovation in organizations is still dependent on heuristic knowledge of employees, and is not based on methods or theories that explain and provide guidance in this process. As a result, the innovation process is frequently considered a black box in which it remains unclear how a certain input results in the observed outcome (Van de Ven and Angle, 2000; Aghion and Tirole, 1994; Fagerberg, 2005).

Innovation can take various forms. In this paper, we are concerned with the improvement of organizational structures to increase the efficiency and effectiveness of organizations. The recent research efforts in the enterprise architecture domain are very relevant in this regard. The ultimate goal of the enterprise architecture domain is to construct organizations that are able to conduct their business in a more efficient and effective manner. Several enterprise architecture frameworks have been proposed in literature that try to make the complexity of organizations more manageable by the use of a systematic approach. Most of these frameworks acknowledge the importance of aligning the IT infrastructure with the enterprise architecture (Zachman, 1987; The Open Group,



Huysmans P., Ven K. and Verelst J. (2010).
DESIGNING FOR INNOVATION - Using Enterprise Ontology Theory to Improve Business-IT Alignment.
In *Proceedings of the Multi-Conference on Innovative Developments in ICT*, pages 177-186
DOI: 10.5220/0003033601770186

2003; Chan et al., 1997). An IT architecture which is aligned with the enterprise architecture contributes to diverse business goals, such as a reduced time to market, the entering of new markets, and support for improved business processes (Kazman and Bass, 2005).

Enterprise architecture frameworks are currently faced with two important challenges. First, a shortcoming of many enterprise architecture frameworks is that they have a descriptive, rather than prescriptive nature (Hoogervorst, 2009). From an innovation management point-of-view, this means that these frameworks are unable to open the black box of the innovation process within the organization. Although such frameworks are able to describe the original and the revised structure of the organization, it remains unclear why the applied changes resulted in a desirable outcome for the organization. This insight is essential to be able to repeat the process in the future. Hence, enterprise architecture frameworks should allow for *repeatability* and *reproducibility*. Repeatability refers to whether the approach would lead to the same results if it was repeated in the same context. Reproducibility refers to whether the approach would lead to the same results if it was repeated in a different context (e.g., in a different organization or with different people).

A second challenge is that organizations are competing in increasingly volatile environments. In such environments, it is important that organizations can quickly adapt to changes in their environment. It has been noted that in such contexts, no long-term competitive advantages can be obtained, and that organizations need to strive towards realizing a succession of short-term competitive advantages (Teece et al., 1997; Eisenhardt and Martin, 2000). Hence, even if organizations succeed to innovate with IT, they will need to ensure that the IT and enterprise architecture is flexible enough to adapt to a changing environment. This requires that models created by enterprise architecture frameworks are *evolvable*. Evolvability is an important property of an architecture. As mentioned by Garlan and Perry: "*software architecture can expose the dimensions along which a system is expected to evolve. By making explicit the load-bearing walls of a system, system maintainers can better understand the ramifications of changes, and thereby more accurately estimate the cost of modifications.*" (Garlan and Perry, 1995).

In this paper, we focus on a case of a European organization that was able to realize substantial business innovation by aligning its IT architecture to its enterprise architecture. Notwithstanding the successful outcome of this enterprise architecture project, the approach taken by the organization strongly relied on the heuristic knowledge of employees, thereby limiting the repeatability and reproducibility of their approach. In addition, it remains unclear whether the modeling technique that was used will be able to provide the required level of evolvability in the future. Given the benefits identified in this case, it seems useful to investigate whether a systematic method could have been used instead, which would allow to recreate these results in other organizations. To this end, we take a design science approach by repeating the enterprise architecture project using a different method. The only systematic method we are aware of that has the potential to address both issues on an enterprise architecture level is Enterprise Ontology. A distinct property of Enterprise Ontology is that it creates models of an organization at the *ontological level* (Dietz, 2006b). As a result, the models are independent from the actual implementation of the processes, thereby providing the required evolvability. In addition, Enterprise Ontology has a strong theoretical foundation that ensures a more exact way of modeling and therefore contributes to the repeatability and reproducibility of the modeling effort.

## 2 METHODOLOGY

Given the focus of our research, we followed the design science research methodology. This methodology is suited to provide the required research setting as it is primarily aimed at *solving* problems by developing and testing *artefacts*, rather than *explaining* them by developing and testing *theoretical hypotheses* (Simon, 1996). In this paper, we focus on an enterprise architecture project undertaken by GFMC (Gas Flow Manager Company). The core business of GFMC is gas transport by handling the delivery of gas across its network. GFMC provided better IT support for business innovation by aligning its enterprise and IT architecture. However, GFMC did not explicitly use a theory or method. Instead, the successful design outcome was the result of previous experience and knowledge of the interaction between the organization and its IT systems. An essential part of design science is the use of relevant theories to develop new solutions for ill-structured problems (Simon, 1996; Klahr and Simon, 1999; Peffers et al., 2007). Repeating the design process using a theoretical foundation is useful in this case, since it aids in achieving similar results in a different context where such experience and knowledge may be lacking. Moreover, the application of a systematic method improves the understanding of the reasons why the use of this particular model achieved such results.

Our design science process is based on the framework proposed by Peffers et al. (Peffers et al., 2007). Based on this framework, we completed three phases. In the first phase, the objectives of the designed solution were identified. Our objective was to design an alternative model of the IT architecture of GFMC that has a strong theoretical foundation. Therefore, a descriptive case study approach was used to study GFMC and its current IT architecture. Using the key informant method, we selected two informants who were highly involved in the development and use of the IT architecture. Our informants were the IT application architect, in charge of the design of the IT architecture, and a business user responsible for formulating changes in business requirements for the IT systems. The primary mode of data collection consisted of a face-to-face interview, which was digitally recorded for future reference. Follow-up questions on the interview took place via e-mail. Additional data sources such as internal documents and presentations, press releases and online information were used to complement the gathered information.

In the second phase, which corresponds to the design and development phase (Peffers et al., 2007), the information gathered in the case study was used to develop a new model. Enterprise Ontology was selected as the underlying theory. For the development of the model, we followed the modeling method as described by Dietz (Dietz, 2006b, p. 138). The initial version of the model took 8 man-hours to complete by the primary author of this paper, after which it was iteratively refined by the different co-authors. The resulting Enterprise Ontology model was then compared to the GFMC model, in order to identify similarities and differences. These differences could occur because of intrinsic differences between characteristics of the modeling languages, or because of a difference in the level of abstraction of both models.

In the third phase, which corresponds to the evaluation phase (Peffers et al., 2007), the characteristics of the developed Enterprise Ontology model were linked to the results achieved by the original GFMC model. For the evaluation, both intrinsic characteristics of the Enterprise Ontology model as well as similarities with the original GFMC model were used as arguments to demonstrate the contributions to business innovation.

## 3 ENTERPRISE ONTOLOGY

In this paper, we use Enterprise Ontology to create enterprise architecture models. Enterprise Ontology is based on scientific theories and offers a systematic method to derive abstract, essential models of an organization. Since Enterprise Ontology considers the organization as a social system, abstraction is guided by the layers defined in communication theory (Stamper et al., 2000). This background ensures that it is well suited to describe the interaction between an organization and its environment (e.g., the market, suppliers, and customers). Enterprise Ontology assumes that communication between human actors is a necessary and sufficient basis for a theory of organizations (Dietz, 2006b). This statement is based on the language action perspective (Flores and Ludlow, 1980) and Habermas' theory of communicative action (Habermas, 1984). The strong theoretical foundation ensures a consistent modeling methodology: since the abstraction method is founded on disjunct theoretical layers (Stamper et al., 2000), the inclusion of a given element in the model does not depend on the interpretation of the modeler. Instead, Enterprise Ontology unambiguously describes which elements should be included, and which elements should be abstracted away. Therefore, for any given situation and scope, it is argued that only one correct Enterprise Ontology model can be constructed (Dietz, 2006b).

Ontological models focus only on *essential* actions. Consider, for example, a shipper who needs to book a certain service with GFMC. If we were to model the request for the service, we could focus on the *form* in which this request occurs. For example, in a multi-channel context, the request could be sent using a web site, a plain e-mail message, or a telephone call. We could also focus on the *information* which is provided with such a request. In that case, the resulting model would contain, for example, the contact details of the shipper, the various selected options, and the required service level. However, the *essential* action is that the shipper requests the booking of a service. Enterprise Ontology models abstract away the form and information aspects, and focus on the ontological actions. Since only the *ontological* actions are represented in the models, the same model will be created for organizations who perform the same function, but operate differently. Therefore, Enterprise Ontology models are *implementation-independent*. For example, consider the BPR case at Ford (Hammer, 1990). The ontological model of the processes of the situation before and after reengineering are identical (Dietz, 2006a). Notwithstanding the lack of change on the ontological level, the concrete implementation of the process had changed drastically. Because of the focus on the essential business processes, Enterprise Ontology models can be very concise. Research has shown that less than 30% of the information related to organizational processes is located on the ontological layer (Dietz, 2006b). Hence, by only focusing on

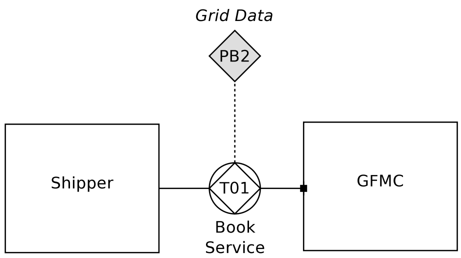

Figure 1: Example Interstriction Model.

the ontological level, model documentation can be reduced by 70%. Therefore, the complete scope of an organization can be represented in a relatively small model. Various case studies are reported where large organizations are described with few modeling artefacts without losing essential or ontological information (Mulder, 2006).

According to Enterprise Ontology, actors perform ontological acts in order to reach a certain goal or result. This result is represented by a *production fact*. The coordination to achieve such a production fact can be described by a universal *transaction pattern*. A transaction always requires the collaboration of two actors: the *initiator* actor who wants to achieve the production fact, and the *executor* actor who performs the necessary actions to create the production fact. The following are examples of production facts which would be achieved by completing a transaction:

- *Delivering a Product.* The initiator actor is a customer, who wants to receive a certain product. The executor is the company which provides the product. A successful transaction would create the fact “Product X has been delivered”.
- *Performing a Service.* Facts do not have to revolve around physical products. In this transaction, the executor could, for example, perform a car maintenance.
- *Subscribing to an Insurance.* In this transaction, the initiator actor would request the insurance of a certain policy. The insurance company is the executor. The completion of this transaction would result in a fact which is required for the ability to make an insurance claim.

The transaction pattern is the most important construct in Enterprise Ontology models. In an Actor-Transaction Diagram (ATD), the transactions are linked to the initiator and executor actors. An Interstriction Model (ISM) adds information-dependencies between transactions. In this paper, we focus on the Interstriction Model. Of all Enterprise Ontology models, the ISM is the most compact (Dietz, 2006b). The clear boundary between the organization, its suppliers and its customers “*makes it preeminently suitable for strategic alignment discussions*” (Dietz, 2006b, p. 170). Moreover, one of its original applications is “*being the background for charting the existing information systems and applications in an enterprise, as the first step in studying the overlap of these systems, and finding the blank spots*” (Dietz, 2006b, p. 213). While other models are available in Enterprise Ontology, we limit ourselves to the ISM since it provides the organizational overview needed for our application. Other models, such as the Process Structure Diagram (PSD), focus on the coordination needed to complete the transaction.

An example of an ISM is shown in Figure 1. A transaction is represented by a diamond enclosed in a circle. The actors are represented by labeled boxes. The initiator actor is connected to the transaction by a line. The executor actor is connected to the transaction by line ending in a black square. If the execution of a transaction depends on information created in other transaction, these transactions are connected by a dotted line. In case such transactions are outside the scope of the current model, they are represented by information banks which represent the results of these transactions. Information banks are depicted as a diamond in an ISM. In Figure 1, a transaction is shown where a shipper issues a request to book a new service with GFMC. In order to be able to complete this request, information about the grid is required from the information bank (PB2).

# 4 FINDINGS

In this section, we will elaborate on the enterprise architecture project undertaken by GFMC. We will first discuss the enterprise architecture effort conducted by the organization and present the model that was created by GFMC. Next, we illustrate the advantages that were realized by GFMC as a result of the better alignment between business and IT and that stimulated innovation. Finally, we present the results of our own enterprise architecture modeling effort using Enterprise Ontology.

## 4.1 Case Description

The enterprise architecture project at GFMC was triggered in 2001 by an important change in legislation. As a result of this change, the organization was forced to separate from a gas trading organization, thereby causing a drastic change in the services offered by the organization. The new core activities are: (1) transmission of gas from the country border to power stations, major industrial end users and distribution system operators (transport); (2) border-to-border transmission of gas destined for other countries (transit); and (3) loading and unloading ships carrying gas (terminalling). The transport of gas is supported by a grid consisting of entry points, nodes, and pipelines connecting the nodes. GFMC offers services based on this grid and negotiates contracts with gas shippers who use the grid. Once a contract has been made, these shippers can submit nominations that indicate the amount of gas they transport over the grid. Based on these nominations, GFMC can plan and execute the gas flow. The actual gas flow is monitored and determines the actual cost allocated to a certain shipper.

As a result of the separation from the gas trading organization and the focus on new services, the entire IT infrastructure supporting the organization had to be rebuilt. Prior to this change, IT was generally considered to be a bottleneck during the implementation of changes by business users. According to our respondents, "*the increasing business complexity and increasing rate of change requests necessitated a new approach*". Moreover, a more structured approach to implementing changes to the IT infrastructure was needed. Given the reputation of IT being a bottleneck for business change, the IT department attempted to implement changes as quickly as possible. Our respondents indicated that this "*resulted in quick fixes which backfired later*". For example, business users once expressed the need to provide a new service with additional options to an existing shipper. The IT department implemented this change in their IT infrastructure by allowing the addition of so-called virtual connection points to the grid. These virtual connections points did not exist in the physical grid, but supported additional features compared to regular connection points. Although this solution worked satisfactory in the beginning, a problem was identified when it turned out to be impossible to neglect these virtual connection points during financial reporting, thereby resulting in incorrect reports.

The new IT system therefore needed to be able to respond better and more quickly to changing business requirements and had to be understandable by business users, so they could more realistically estimate the impact of a requested change. It was decided to align the IT architecture with a high-level enterprise architecture model. This approach was inspired by the Service-Oriented Architecture (SOA) paradigm. In SOA, services are IT artefacts which correspond to concepts used in the business context (Erl, 2005). It has therefore been argued that the use of SOA can contribute to a better alignment between business and

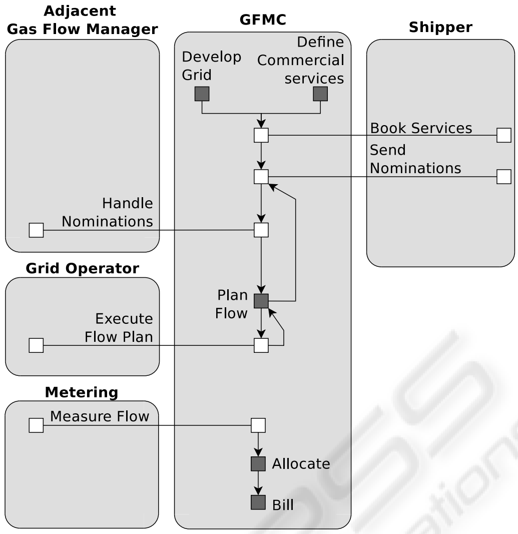

Figure 2: Role Activity Diagram.

IT (Erl, 2005). It is claimed that business agility can be achieved by "*assembling new business processes from existing services to meet changing marketplace needs*" (Dan et al., 2008). This view would allow business users to implement changes themselves, by differently orchestrating services. However, research shows that SOA adoption approaches vary, and that the selected approach impacts the resulting value (Chua, 2009). It should be noted that GFMC did not attempt to completely adhere to the SOA paradigm, or implement a system consisting only of services. The main idea that was adopted from the SOA paradigm was to align the artefacts which make up the IT architecture with concepts from the business context.

The high-level enterprise architecture model on which the IT architecture would be based was constructed using a Role-Activity Diagram (RAD). The RAD developed by GFMC is shown in Figure 2. The model only contains business activities that describe the high-level operation of GFMC. In a RAD, activities are modeled within the responsible organizational entity (actor roles). Depending on the scope of the model, these organizational entities can represent organizations, departments or actual employees. In this case, an abstract RAD was used, describing the interaction of the organization with its partners. An abstract RAD model describes how the organization works, without detailing how the specific activities are implemented (Ould, 2005, p. 234). Internally executed activities are represented by gray boxes. Activities which require collaboration with external partners are represented using white boxes in the collaborating entities, which are connected by a solid line. The arrows represent the process flow. The concepts used in the RAD were defined in a glossary, which was developed iteratively with business users. This was necessary to make sure that the model was interpreted consistently across the organization. Moreover, this ensured that business users understood at least the concepts in the model. Based on this model, the IT application portfolio was constructed. Every activity of the RAD which is carried out by GFMC was supported by exactly one application. The scope of these applications was defined so that (a) their functionality did not overlap, and (b) their combined functionality would support the whole RAD. Moreover, the applications needed to be as independent from each other as possible in order to be able to implement changes in one application without affecting another. This is consistent with the concept of *loose coupling* in SOA: in order to be able to orchestrate services differently, they need to be independent from each other. According to our respondents, the common definition of concepts as captured by the glossary was a necessary precondition to build cooperating but independent applications.

## 4.2 Realized Advantages

This approach yielded several successful results. While many technical improvements have been achieved, we elaborate mainly on the improvements that have a business-related impact, and on IT-related improvements which result from the use of the business model.

**Alignment.** The abstract enterprise architecture model aided in better aligning the IT applications with the business activities for multiple reasons. First, the glossary ensured that IT staff and business users shared the same concepts, and had a common understanding of these concepts. The glossary was used to identify the underlying data model on which the applications operate. Since the glossary was developed in cooperation with business people, the important data elements should reflect the information used by business users. Second, the model itself can be understood by business users since they recognize the actors as the parties with whom they interact. Since the applications are based on the activities in the RAD, business users also understand the scope of the applications. Moreover, these factors aid communication. Our respondents report that "*discussions concerning change requests are much more structured and efficient*".

**Change Assessment.** Since business users understand the RAD, they also understand the way the applications interact with each other. Therefore, they can assess the impact of a change they propose to the IT system. When a business user who is responsible for an activity in the RAD requests a change, it is possible that this change will affect other applications. When the scope of a change is related to multiple activities in the RAD, it is straightforward for the business user to understand which RAD activities are affected, and therefore which departments should be involved in the project. Moreover, when many applications are affected by the change, a phased implementation may be necessary. Our respondents reported that "*thanks to the model, business users better understand why and how we use a phased implementation*".

**Reuse.** According to our respondents, it was essential that no redundant functionality was allowed in order to stimulate reuse. This ensures that required functionality that is not in scope for a certain application is reused from other applications. For example, when a service is requested in the activity *Book Services* on a connection point which is not present in the grid, it cannot be created in the *Book Services*-application. Instead, a business rule will check whether the needed connection points exist, and if this is not case, the *Develop Grid*-application will be used to create the connection point. This separation of functionality ensures that changes in functionality have to be implemented in only one application. The RAD model also allowed to identify common functionality in the three activity areas of the organization (i.e., transport, transit and terminalling). Currently, overlapping functionality in different domains is supported by applications that share the same code base, but operate on different databases. Only applications that are domain-specific are developed separately.

**Development Process.** Given the scope limitations of the applications and the desire to keep them independent, a more structured development process was needed. IT staff preferred to bundle changes in a fixed release cycle, in order to be able to test the impact of these changes beforehand. However, this was considered as an important constraint by the business, since it delayed the implementation. Since business users now understand and support the need to adhere to the IT architecture, they accept that changes are implemented during fixed release cycles. Prior to the alignment, business users demanded that changes were implemented as soon as possible, which could have negative effects in the long term (cf., the virtual connection points example). Moreover, testing application changes improved in the new IT environment. Since the applications are loosely coupled, the impact of changes in a certain application is limited to that application. When coupling is needed, interfaces are used. Therefore, changes regarding the implementation of the interface are hidden from other applications. Aligning the applications with business functionality also proved beneficial in other development areas. Our informants mentioned that "*the RAD clearly showed a business chunk which was not supported by an application. We needed to integrate that specific business part with our ERP system. However, this was very cumbersome, since no data concerning that part was available. Now that we identified this gap, and developed an application, the integration is much easier.*" This example suggests that an abstract business model aids the completeness of the IT infrastructure.

## 4.3 Enterprise Ontology Model

In this section, we present the model which was developed using the Enterprise Ontology theory, and compare it to the RAD model developed by GFMC. The Enterprise Ontology model was developed by the authors of this paper, in order to ensure the correct application of the Enterprise Ontology methodology, and to eliminate the influence of the heuristic knowledge of the GFMC employees. We chose not to start from the existing RAD model, but to base ourselves on the information and insights gathered in the descriptive case study. Based on this data, we identified relevant production facts. These included the developed grid, the offered services, the nominations, the flow plan, the measurements, and the allocations. In Enterprise Ontology, exactly one executor actor is responsible for the creation of a production fact. Therefore, two distinct transactions are needed to describe the acts concerning the flow plan: GFMC creates the flow plan itself, but the physical execution is handled by an external actor. The same argument applies to the allocations: while GFMC calculates the allocations, it expects their clients to pay them. Therefore, separate allocate and bill transactions are required. The resulting ISM model from Enterprise Ontology is shown in Figure 3.

When comparing this model to the RAD developed by GFMC, it should be noted that both models are very similar. The actors—which were important for business users to recognize—are identical in both models. The transactions in the Enterprise Ontology model correspond largely to the activities in the RAD. Only the RAD activity *Define Commercial Services* is represented in the Enterprise Ontology model as an information bank, since the list of commercial services offered by GFMC is considered to be stable. The development of new services was therefore not within the scope of the ISM model. However, as a

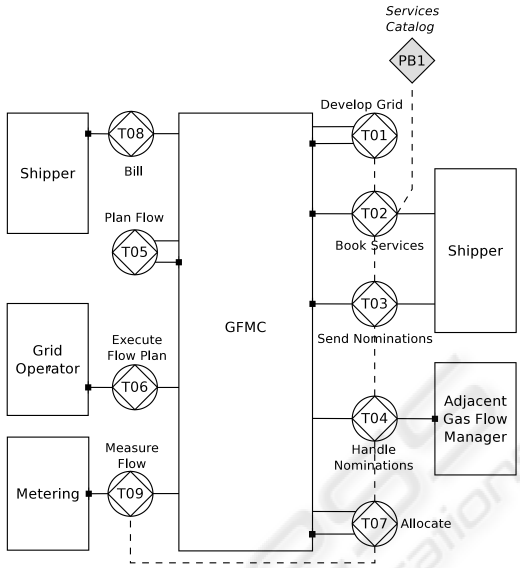


Figure 3: Interstriction Model for GFMC.

result of the different modeling languages, some differences can be noted. On the one hand, the process flow between the activities is represented in the RAD, while it is missing in the ISM. In Enterprise Ontology, the process flow is represented in the Process Structure Diagram, which is outside the scope of this paper. On the other hand, the ISM represents the information dependencies between transactions, which cannot be derived from the RAD. Since the applications in GFMC are based on the RAD activities, these dependencies provide useful information concerning the coupling of these applications.

Given the similarities between the RAD and the ISM, we can expect that the use of a structured modeling approach based on Enterprise Ontology would have delivered the same results for GFMC. Previous research indeed indicates that Enterprise Ontology models can improve evolvability and support innovation. First, Enterprise Ontology models enable communication between stakeholders who are involved in organizational changes. Since Enterprise Ontology models are very concise, they are easier to understand than elaborate, detailed models. Studies show that executives can understand and reason using these models (Mulder, 2006). Second, the integration of production facts with their coordination acts improves modularity. Because of the cohesiveness of transactions, it is clear which parts of the current organizational implementation depend on each other. For example, Op't Land showed that Enterprise Ontology can be used for splitting organizations (Op 't Land, 2008). Based on Enterprise Ontology models, suggestions for splitting an organization could be made which correlate with decisions taken by top managers based on their heuristic knowledge. Third, Enterprise Ontology provides a *constructional view* of the organization, instead of a *functional view* (Dietz, 2006b). A constructional view focuses on *how* an organization works, instead of describing what its function is. Therefore, interactions among the components of the organizational model can be identified in Enterprise Ontology models, such as the information dependencies. Such interactions provide insight concerning the impact of a change. Providing this insight has also proved to be a valuable characteristic in the case study.

## 5 DISCUSSION

The case of GFMC provides a good example of how IT can deliver value for the organization and support innovation. With the new IT architecture, GFMC was able to increase alignment between the enterprise and IT architecture by making sure that both parties understand each other and share the same goals. In addition, the efficiency of the organization could be improved by removing redundant parts in the software architecture and by making sure that all parts of the organization are properly supported by IT. The transformation in GFMC was enabled by the heuristic knowledge and experience of employees. It is commonly known that experienced and highly-qualified professionals are able to use their heuristic knowledge to create models that provide a high-quality solution for organizations. As a result, much problem-solving in organizations is based on heuristic knowledge held by employees. Current methods are not able to provide a fully-fledged alternative for this heuristic knowledge. This is actually one reason why methodologies in information systems development are seldom used by experienced developers: methodologies provide useful background knowledge for junior developers, while more experienced developers feel constrained by methodologies and prefer to rely on their own experience (Fitzgerald, 1998). Heuristics are considered to provide solutions that are of a sufficiently high quality and that cannot be improved by using methodologies.

Although the solution devised by GFMC has resulted in measurable advantages, this solution is still based on the heuristic knowledge of its employees. The organization is in that case dependent on the knowledge and expertise possessed by its employees. As previously argued, it is desirable from an innovation management perspective to design a method that has the potential to ensure repeatability and reproducibility. Otherwise, an organization will be dependent on the characteristics of specific persons in the organization for innovation. This represents a situation in which there is a lack of a well-defined and standardized process which is difficult to manage. Such situations are associated with the lower levels in maturity models such as CMMI. If some level of repeatability and reproducibility were to be realized, it would facilitate innovation management. Evidently, this does not guarantee that the organization can produce one innovation after the other. However, the organization will better understand its innovation process and will be better able to manage and guide this process.

In this paper, we have shown that Enterprise Ontology has the potential to assist organizations in enabling change and innovation in organizations. A first important advantage of Enterprise Ontology is that it does not rely on heuristic knowledge, but provides a theoretical foundation and a clear description of a method to develop organizational models. This allows Enterprise Ontology to provide the required repeatability and reproducibility in its approach. It was observed that the ISM was very similar to the RAD model. This is actually very desirable since the RAD model has resulted in significant advantages for the organization. If the ISM had been significantly different, then it would have been unclear whether the ISM also would have had the potential to realize the same advantages. This indicates that the Enterprise Ontology methodology could be a valid candidate to serve as the basis of an innovation management method. The main advantage of Enterprise Ontology is that it provides a theoretical foundation for constructing models that eliminate reliance upon heuristic knowledge, design decisions of the modeler, and the chosen level of abstraction. Intuitively, the GFMC employees created the RAD model by only modeling the ontological processes in the organization. However, this decision was solely inspired by their heuristic knowledge and experience, and was not enforced by their methodology. Indeed, many modeling techniques—including RAD—allow the modeler to decide on the desired level of abstraction. This implies that different models can be drawn for the same situation, depending on the aim and the experience of the modeler. This therefore creates many degrees-of-freedom and does not allow to make the process repeatable or reproducible. Enterprise Ontology, however, ensures that only the ontological transactions within the organization are included in a model, and therefore enforces that a single level of abstraction is used. Thanks to this property, it is claimed that only one correct Enterprise Ontology model can be created for a specific situation (Dietz, 2006b), thereby reducing the degrees-of-freedom for the modeler. This further enables Enterprise Ontology to provide a repeatable and reproducible approach in the design of organizations. It is important to note that we do not claim that the ISM is inherently better than the RAD model; rather we emphasize that Enterprise Ontology has more potential for realizing repeatability and reproducibility than RAD, which are important properties of a method to improve innovation management.

A second important advantage of Enterprise Ontology is that its ontological approach allows to increase the flexibility and evolvability of the organization. The ISM showed that each transaction identified by Enterprise Ontology was translated into an application by GFMC. As mentioned earlier, a change on the ontological level reflects an essential change in the business process of the organization. Such changes will definitely have an impact on the information systems that support the business process. In this case, a change on the ontological level can be translated into a change to an existing application, the addition of a new application, the deletion of an existing application, or a combination of these. This provides traceability between the enterprise architecture and software architecture. Moreover, the underlying software architecture can change independently from the enterprise architecture to improve the efficiency or service quality of the information systems. Since ontological models neglect the specific implementation, they remain correct when changes are applied to the current implementation. Therefore, such changes are not considered to be innovative on the ontological level. However, as shown by the benefits in the GFMC case, many improvements are possible in the implementation of a process. In such cases, a stable ontological model aids change assesment, stimulates reuse, and prevents violations against the achieved alignment. Although the implementation in GFMC was inspired by SOA as the implementation technology, this is not the only potential technology option. In ongoing research, we are working on translating the Enterprise Ontology level to the software level to obtain traceability between between both levels and to increase the evolvability of information systems (Van Nuffel et al., 2010; Huysmans et al., 2010). Hence, although Enterprise Ontology provides high-level models, they have the potential to be aligned with the underlying software architecture, thereby providing traceability and evolvability.

## 6 CONCLUSIONS

In this paper, we have presented a case of an organization that realized business innovation by obtaining a better alignment of its enterprise and IT infrastructure. This alignment resulted in substantial advantages for the organization. Given this success, we were concerned with how other organizations can recreate these advantages so that managers are given more insight into the innovation process. Our case study showed, however, that the organization strongly relied on the heuristic knowledge of their employees. Therefore, we repeated the modeling effort by using a more systematic method that has a strong theoretical foundation. The Enterprise Ontology theory was used to create an ontological model of the organization. Our results showed that the Enterprise Ontology model was very similar to the enterprise model developed by the organization. It therefore seems reasonable to expect that the Enterprise Ontology model would have been able to realize the same advantages in the organization. The main advantage of Enterprise Ontology is that it provides a more repeatable and reproducible result and that the resulting models are more evolvable.

Our paper has two main contributions. First, we have demonstrated that the use of Enterprise Ontology resulted in the a very similar enterprise architecture model as the model developed by the organization. The use of Enterprise Ontology is a definite improvement over the reliance on the heuristic knowledge of employees, since it increases the understanding of the innovation process using a theoretical foundation. Second, we have shown that Enterprise Ontology can be used as a starting point to derive the IT architecture of an organization. This increases alignment between the enterprise and IT architecture.

A limitation of this study is the use of a single case which limits its external validity. Hence, we do not claim that following the Enterprise Ontology theory will always result in innovation in the organization. Nevertheless, Enterprise Ontology seems a promising approach to improve business-IT alignment in organizations which has been known to lead to several advantages. It therefore appears that Enterprise Ontology has the potential to realize IT-enabled innovation. Our ongoing research efforts therefore focus on devising a method to directly translate Enterprise Ontology models to the IT infrastructure of organizations to improve their flexibility.

## REFERENCES

Aghion, P. and Tirole, J. (1994). Opening the black box of innovation. *European Economic Review*, 38(3/4):701–710.

Brynjolfsson, E. and Saunders, A. (2010). *Wired for Inno-*

*vation: How Information Technology is Reshaping the Economy*. The MIT Press, Cambridge, MA.

Chan, Y. E., Huff, S. L., Barclay, D. W., and Copeland, D. G. (1997). Business strategic orientation, information systems strategic orientation, and strategic alignment. *Information Systems Research*, 8(2):125.

Chua, F.-F. (2009). Adoption of service-oriented architecture by information systems. *International Journal of Autonomous and Adaptive Communications Systems*, 2(4):317 – 330.

Dan, A., Johnson, R. D., and Carrato, T. (2008). Soa service reuse by design. In *SDSOA '08: Proceedings of the 2nd international workshop on Systems development in SOA environments*, pages 25–28, New York, NY, USA. ACM.

Dietz, J. L. (2006a). The deep structure of business processes. *Communications of the ACM*, 49(5):58–64.

Dietz, J. L. (2006b). *Enterprise Ontology: Theory and Methodology*. Springer, Berlin.

Eisenhardt, K. M. and Martin, J. A. (2000). Dynamic capabilities: What are they? *Strategic Management Journal*, 21(10/11):1105–1121.

Erl, T. (2005). *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Fagerberg, J. (2005). Innovation: A guide to the literature. In Fagerberg, J., Mowery, D. C., and Nelson, R. R., editors, *The Oxford handbook of innovation*. Oxford University Press, New York, NY.

Fitzgerald, B. (1998). An empirical investigation into the adoption of systems development methodologies. *Information & Management*, 34(6):317–328.

Flores, F. and Ludlow, J. (1980). Doing and speaking in the office. In Fick, G. and Sprague, R. H., editors, *Decision Support Systems: Issues and Challenges*, pages 95–118. Pergamon Press, New York, NY.

Garlan, D. and Perry, D. E. (1995). Introduction to the special issue on software architecture. *IEEE Transactions on Software Engineering*, 21(4):269–274.

Habermas, J. (1984). *The Theory of Communicative Action: Reason and Rationalization of Society*, volume 1. Beacon Press, Boston, MA.

Hammer, M. (1990). Reengineering work: Don't automate, obliterate. *Harvard Business Review*, 68(4):104ff.

Hoogervorst, J. A. (2009). *Enterprise Governance and Enterprise Engineering*. The Enterprise Engineering Series. Springer, Berlin.

Huysmans, P., Bellens, D., Van Nuffel, D., and Ven, K. (2010). Aligning the constructs of enterprise ontology and normalized systems. In *Proceedings of the 6th International Workshop on Cooperation & Interoperability – Architecture & Ontology (CIAO! 2010), June 4–5, 2010, St. Gallen, Switzerland*.

Kazman, R. and Bass, L. (2005). Categorizing business goals for software architectures. Technical report, Software Engineering Institute. CMU/SEI-2005-TR-021.

Klahr, D. and Simon, H. A. (1999). Studies of scientific discovery: Complementary approaches and convergent findings. *Psychological Bulletin*, 125(5):524–543.

Mulder, H. (2006). *Rapid Enterprise Design*. PhD thesis, TU Delft.

Op 't Land, M. (2008). *Applying Architecture and Ontology to the Splitting and Allying of Enterprises*. PhD thesis, TU Delft.

Ould, M. (2005). *Business Process Management, a Rigorous Approach*. The British Computer Society, Swindon, UK.

Peffers, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3):45–77.

Simon, H. A. (1996). *The Sciences of the Artificial*. MIT Press, Cambridge, Massachusetts, third edition.

Stamper, R., Liu, K., Hafkamp, M., and Ades, Y. (2000). Understanding the roles of signs and norms in organizations. *Journal of Behavior and Information Technology*.

Teece, D. J., editor (1987). *The Competitive Challenge: Strategies for Industrial Innovation and Renewal*. Ballinger Publishing Company, Cambridge, MA.

Teece, D. J., Pisano, G., and Shuen, A. (1997). Dynamic capabilities and strategic management. *Strategic Management Journal*, 18(7):509–533.

The Open Group (2003). The open group architecture framework (togaf) version 8.1. Technical report.

Van de Ven, A. H. and Angle, H. L. (2000). An introduction to the minnesota innovation research program. In Van de Ven, A. H., Angle, H. L., and Poole, M. S., editors, *Research on the Management of Innovation: The Minnesota Studies*. Oxford University Press, New York, NY.

Van Nuffel, D., Huysmans, P., Bellens, D., and Ven, K. (2010). Towards deterministically constructing organizations based on the normalized systems approach. In *Proceedings of the 5th International Conference on Design Science Research in Information Systems and Technology (DESRIST 2010), June 4–5, 2010, St. Gallen, Switzerland*.

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