A PHILOSOPHICAL FOUNDATION FOR BUSINESS AND IT ALIGNMENT IN ENTERPRISE ARCHITECTURE WITH THE EXAMPLE OF SEAM

Gil Regev, Biljana Bajić-Bizumić, Arash Golnam, George Popescu,
Gorica Tapandjieva, Anshuman B. Saxena, Alain Wegmann
École Polytechnique Fédérale de Lausanne (EPFL),
School of Computer and Communication Sciences
CH-1015 Lausanne, Switzerland
{gil.regev, biljana.bajic, arash.golnam, george.popescu, gorica.tapandjieva, anshuman.saxena, alain.wegmann}@epfl.ch

Keywords: Business-IT Alignment, Theory, Philosophy, Methodology, SEAM, Service, Systems Inquiry, PhD Hiring.

Abstract: Business-IT alignment is complicated because of the need to align multiple business and IT points of view. A philosophical foundation can help generate methods that bring together these disparate viewpoints in a common model that all stakeholders can agree upon. In this paper, we describe the philosophical foundations of the Systemic Enterprise Architecture Methodology (SEAM) and show how it can help business-IT alignment with the example of a concrete business process. These foundations are applicable to other methods as well.

1 INTRODUCTION

The subject of business and IT alignment has been the focus of intensive research for over twenty years: see for example (Chan and Reich, 2007). It has also been a major concern for IT executives (Luftman and McLean, 2005). During all this time, it seems that few (if any) methods with a theoretical grounding have been proposed by researchers in this field (Chan and Reich, 2007). This is the more surprising as it has been noted that cultural issues may be at the heart of misalignment between business and IT and that, despite the general tendency to believe otherwise, misalignment may not be counterproductive to some firms (Chan and Reich, 2007).

For many years, we have been contributing to the business and IT alignment field by building and applying an Enterprise Architecture method called SEAM. SEAM has an explicit theoretical grounding, or more precisely a philosophical grounding, which we describe in this paper.

Enterprise Architecture (Zachman, 1987) was created in the late 1980s in order to help IT departments to design IT systems that support the increasing complexity of businesses. This attempt was based on the premise that businesses increasingly depend on their IT systems, and that these systems "keep the business from disintegrating" (Zachman, 1987). The term Enterprise Architecture (EA), initially referred to as information systems architecture, reflects this understanding that the information systems of an organization mirror the business itself. This has resulted in research into the combined fields of Enterprise Architecture and Business-IT alignment.

Many EA frameworks have been proposed since then. For example TOGAF (The Open Group, 2009) and ArchiMate (Lankhorst et al., 2009). In general these frameworks have no explicit theoretical grounding. They are implicitly based on strategic management practices that view the enterprise as a machine where executives set the vision and goals that are then refined into IT architecture.

The Zachman framework stands out as having an epistemology in the sense that it has an ontology based on the work of a building architect including a different language for each trade.

SEAM focuses mainly on the enterprise architects' role in helping with business-IT alignment and less on their role in mapping the IT infrastructure.

The term business-IT alignment hides much complexity. In any organization there are indeed many businesses, such as, groups of people, departments, business units, project teams, etc. Each one is a business within a greater business with its own identity, worldview, behavior and structure. IT systems reflect the complexity of their environment. Embarking on business-IT alignment in order to embed this complexity in an IT system is a major challenge. It requires methods that enable enterprise architects to understand the multiple viewpoints, desires and needs of these businesses within the enterprise as well as their external stakeholders. To appreciate and reconcile these points of view, we need to understand what is a business entity and how it sees itself and the world around it. Current Enterprise Architecture methods do not delve sufficiently on these issues.

One of the main concepts used in EA discourse is the "system". Lankhorst et al., for instance, give the examples of large systems such as enterprise information system and software system (Lankhorst et al., 2009). They further note that an architectural approach is needed to manage the complexity of such large systems. General Systems Theory (GST), (von Bertalanffy, 1968) also often called General Systems Thinking (Weinberg, 1975), was designed long ago to provide just the kind of architectural principles. GST can provide theoretical grounding and guide architects of large systems.

SEAM is an EA method that was created from the ground up based on GST. One of the main contributions of SEAM to EA is its reliance on an explicit systemic modeling paradigm (Wegmann, 2003). This paradigm provides a comprehensive explanation of SEAM in terms of its theory, philosophy and methodology. More specifically it provides a way to understand the often disparate viewpoints of the multiple businesses and IT within the organization.

In this paper we provide a complete explanation of the paradigm. We explain how it can be useful in EA by showing its application in SEAM. We provide a short example of SEAM modeling based on a real university process: the hiring of PhD students at EPFL. SEAM is currently used as a modeling method for the EPFL IT organization.

The paper is structured as follows: In Section 2 we present some background on business-IT alignment and EA. In Section 3 we describe the systemic modeling paradigm. In Section 4 we show the application of the paradigm to EA with the example of SEAM. In Section 5 we explain how the use of SEAM for the example of the PhD hiring process at EPFL illustrates the systemic modeling paradigm. In the last section we list some recommendations for future research in EA.

2 SYSTEMIC MODELING PARADIGM

Banathy and Jenlink (Banathy and Jenlink, 2004), seeking to provide a comprehensive description of GST, explain it as the interlinked association of three domains of inquiry: systems theory, systems philosophy (which further contains epistemology, ontology and axiology) and systems methodology (see Figure 1). They call this set Systems Inquiry.

Note that Banathy and Jenlink use the term ontology in its philosophical sense of what the real world contains. In the EA world ontology is more often used in its computer and information sciences meaning of "a set of representational primitives with which to model a domain of knowledge or discourse" (Gruber, 2009).

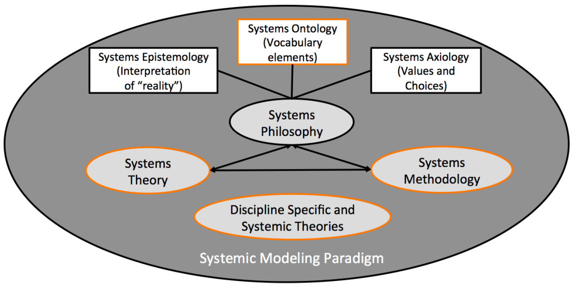

Figure 1: The Systemic Modeling paradigm (expanded from Systems Inquiry).

The systemic modeling paradigm was proposed by Wegmann in (Wegmann, 2003). It combines Systems Inquiry and Kühn's notion of paradigm change. A paradigm is defined as "a philosophical and theoretical framework of a scientific school or discipline within which theories, laws, and generalizations and the experiments performed in support of them are formulated" (Merriam-Webster, 2013). The systemic modeling paradigm also extends Systems Inquiry with discipline specific theories.

2.1 Systems Theory

Systems theory, as described by Banathy and Jenlink (Banathy and Jenlink, 2004) espouses the view that modern science and industry have locked themselves in a pursuit of an "ever-increasing specialization." This specialization results in specialists' the inability, and often unwillingness, to engage with, or even understand, other specialists.

The early system thinkers have observed that as each specialized discipline creates its own specialized vocabulary, it nevertheless uses concepts that are similar to other disciplines. It is often the vocabulary that is different but the underlying principles are the same. The same phenomena studied by a biologist can be observed in enterprises, for example. GST was therefore designed as a *lingua franca* that would enable specialists from different disciplines to collaborate (e.g. a biologist with an economist) and understand each other. GST seeks to define general principles that can be applied to any phenomena across established disciplines, thereby complementing the specialist view.

In addition, Wegmann proposed to use discipline specific theories to complement the general principles offered by the General Systems Theory (Wegmann, 2003).

2.2 Systems Philosophy

As noted by Banathy and Jenlink (Banathy and Jenlink, 2004), the interest of GST with general principles that transcend disciplines implies a close link with philosophy. They define systems philosophy as consisting of three components, Ontology, Epistemology and Axiology (Ethics). Ontology describes what things are, e.g. what a person is, what an organization is, what a society is. Epistemology is oriented towards the questioning of ontology, e.g. how we know what is person, an organization, or society? Banathy and Jenlink contend that these two aspects are intimately linked because it is often impossible to completely separate what we know from how we know it. Finally, axiology is concerned with the notions of value, ethics and aesthetics. It underlines the choices made by systems thinkers when they select some aspects of reality for attention, rather than others. Are these choices good, bad, beautiful, ugly, moral or not, constitute the questions that axiology aims to reply to.

2.3 Systems Methodology

Systems methodology is the study and creation of methods for intervention. Banathy and Jenlink (Banathy and Jenlink 2004) divide systems methodology into two domains of inquiry: the study of methods (their creation and improvement) and the practical use of these methods. The methods are used for the analysis of systems and systems problems, the design, development and implementation of systems and the management of systems in general. The method depends on the problem context and content as well as the type of systems in which the problem is situated. A specific methodology needs to be chosen from the wide range of available frameworks using a solid justification and analysis of the investigated problem.

3 THE SYSTEMIC MODELING PARADIGM APPLIED TO SEAM

Having briefly introduced the systemic modeling paradigm, we now use it to explain how an EA method, such as SEAM, can benefit from this grounding.

3.1 SEAM Systems Theory

SEAM is a method built on a systemic grounding. Much like GST is interested in federating scientific disciplines, when intervening in organizations, there is a need to understand and transcend the specialist view of the stakeholders (often called "silos" today) that compose the organization. While doing so, the enterprise architect should be careful not to alter too much the stakeholders' way of working because their effective action depends on them remaining specialists.

In addition to GST, discipline specific theories can be used as well. These theories can be specific to the discipline of each stakeholder involved, e.g. marketing, sales and software engineering. For example, the theories specific to SEAM are: refinement theory to verify business-IT alignment, first order logic to formalize beliefs and operational semantics to formalize behavior.

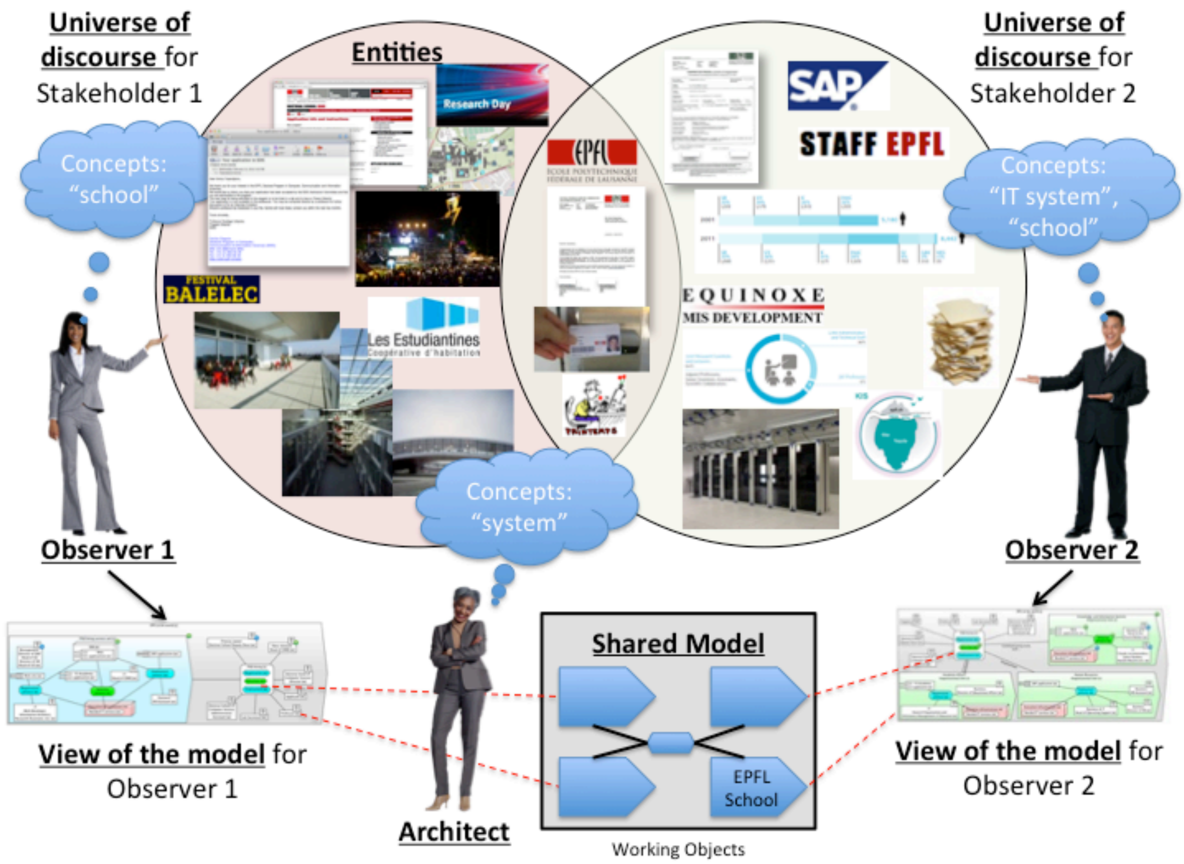

Figure 2: Illustration of the Systemic Modeling Paradigm with three stakeholders: two observers and one architect

3.2 SEAM Systems Philosophy

Parting from Banathy and Jenlink's explanation we explain the SEAM philosophy staring from the epistemology rather than the ontology.

The *SEAM epistemology* (shown in Figure 2) is interpretative (Mintzberg et al., 1998) or interpretive (Checkland and Holwell, 1998). This means that we believe that each stakeholder creates his specialized knowledge of his work by interacting with the work artifacts and through his relationships with other specialists in his domain.

We call universe of discourse this set of entities that the stakeholder sees, which is a subset of the total number of entities available in reality. Two universes of discourse are shown in Figure 2, one for each stakeholder. The enterprise architect is also a specialist who constructs her models from her relationship with stakeholders and other enterprise architects. The universe of discourse of the enterprise architect is implicitly shown. It is composed of the two stakeholders' models. The enterprise architect helps the stakeholders to express their knowledge about their work in a model that can be merged with other stakeholders' models.

Each stakeholder builds a set of concepts, which we call his conceptualization, by interacting with his universe of discourse. This conceptualization is the basis of his understanding of the world. Three conceptualizations are shown as clouds in Figure 2, one for each observer and one for the enterprise architect. The enterprise architect constructs her conceptualization based on the set of stakeholders' conceptualizations.

Other terms that convey a similar meaning to the universe of discourse and conceptualization can be found in Vickers's appreciative system (Vickers, 1968; Regev et al., 2011). Vickers explains that people and organizations develop a readiness to see some aspects of reality. This readiness is necessary for effective action, but is also a barrier to collaboration with others because it makes it difficult to see things from a distinct point of view.

What we call the *SEAM ontology*, in-line with the standard use of the term ontology in computer and information sciences (Gruber, 2009), is the model elements with which an enterprise architect describes the stakeholders' conceptualizations and the shared model that the stakeholders should agree about.

In the SEAM ontology we use the term *working object* to designate a system in the conceptualization. For example, a working object named "EPFL School" in the model maps to a system that the modeler understands as being a school in the conceptualization. The name EPFL helps mapping to the specific school "EPFL" in the universe of discourse. This explains how the model element in the model relates to entities in the universe of discourse.

The ontology in the form of the working object allows benefiting from the domain specific theories proper to SEAM (e.g., refinement, model checking). A working object refers to a service system (Vargo et al, 2008; Regev et al., 2011) in the sense that it shows the way value is co-created rather than an organizational entity, such as a company. The working object "EPFL School" may therefore contain other working objects that map to organizations that most stakeholders will think of as external to EPFL, for example, an IT supplier. Having the "IT supplier" working object within the "EPFL School" working object shows that the service provided by the EPFL School includes the service provided by the IT supplier.

The *SEAM axiology* refers to the choices the specialists make about what to include in their model. These choices can have two aspects: aesthetics and ethics (Lemos 1999). Aesthetics include practicality and simplicity. The modeler needs to decide to model what is useful and practical to show the problems and the possible solutions. The goal is not to make an exhaustive universal list of what exists in a company, but rather to analyze a concrete challenge. The modeler needs also to find a way to attain simplicity. The modeler should use the abstraction mechanisms of SEAM to illustrate concisely the situation. Even if it is concise, the model should keep the important systemic model elements (such as service system boundaries in the "to-be" model), so that the stakeholder can understand what is represented. Ethics – the model captures also the ethical choices of the modeled enterprise. For example, is the shareholder the primary "customer" of the company or should it be the "normal" customer. Axiology is useful to explain these two kinds of choices. It is associated with heuristics, such as, for example, that it is usually beneficial to first understand the "real" customer rather than the shareholder.

3.3 SEAM Systems Methodology

The SEAM methodology prescribes the way an enterprise architect uses the SEAM theory and philosophy to produce results. The methodology is a collection of techniques, some of which are well known to enterprise architects (such as the as-is and to-be modeling). Others were imported from other disciplines, e.g. contextual inquiry (Beyer and Holtzblatt, 1998). Because it is often costly and time consuming to do contextual inquiry in practice, we use an alternative technique of using concrete names of people and organizations (e.g., EPFL School rather than simply School) as well as anecdotes in workshops. This helps stakeholders remember the context they were in when facing some problems. Without this context, they may often forget to give many details about their work. A related technique encouraged in SEAM is to collect supporting evidence about concrete situations in the form of e.g., pictures, letters, and emails.

We also recommend developing a model bottom-up and top-down at the same time. We obtain the best results when the modeling sessions are short and iterative.

A few techniques were extended from standard techniques, e.g. the blackbox-whitebox technique is used to represent systems structure as is customary in engineering, but also to represent the structure of behavior, which is less frequent.

4 EXAMPLE OF THE PHD HIRING PROCESS WORKSHOP

In this chapter we show the importance of the SEAM philosophical grounding with a concrete and real example. We use the results of a one-day business-IT alignment workshop done in Fall 2012 at École Polytechnique Fédérale de Lausanne (EPFL). We illustrate the relation between the SEAM systemic paradigm and the workshop practice. EPFL uses a service-oriented strategy and is currently testing SEAM as a modeling technique to represent its IT services. A workshop was planned to train IT managers for the use of SEAM in order to enable them to model their own services. The workshop was organized by the Laboratory of Systemic Modeling (LAMS) at the request of the IT governance head of EPFL. It was decided to work on the PhD hiring process as an example of a process that involves many departments and IT systems. The PhD hiring process is a good example because it brings together many actors across EPFL with many viewpoints that need to be reconciliated. It was also selected because it is an important process, with no projects currently planned to analyze it. It was therefore “neutral territory”.

4.1 Organizational Description

EPFL is a polytechnic university located in Lausanne, Switzerland. It is organized into seven schools, which are themselves formed of research and teaching units. For the academic year of 2011-2012, EPFL had approx. 8’500 students, including 2000 PhD students. Some 500 new PhD students are hired each year. EPFL has about 4’500 employees.

IT is distributed across the whole EPFL organization. Approximately 80 people work in central services, under direct supervision of the Chief Information Officer (CIO). 20 people work in central services, outside of the CIO supervision. These 20 people manage mostly SAP and the academic management system, called ISA. Some 150 people work in the IT groups attached to the seven schools, or are dedicated to the IT of research and teaching units.

Overall, the IT people manage more than 125 central software applications, e.g. SAP for HR and finance, ISA, as well as some scientific infrastructure such as super-computers.

This distributed nature of the business and IT organizations leads to the co-existence of many viewpoints on any single process. There is a need to federate these viewpoints to improve business and IT alignment.

4.2 Description of the current PhD Hiring Process

The process includes the following 3 phases:

Registration. The registration begins when an applicant fills an application record in ISA. The doctoral program committee analyzes all application records and decides who is admissible to the program. The doctoral program assistant informs, by e-mail, the applicant that he or she is admitted or rejected. The doctoral program assistant also informs by e-mail the professors that the list of admitted applicants is available in ISA.

Selection. The professor organizes interviews with potentially interesting admitted applicants. If the professor and the applicant agree to work together, the offer is formalized in an admission letter signed by the professor and by the doctoral program director. The letter is sent to the applicant. No specific IT system supports this part of the process. It is implemented via e-mails, Word and Excel documents.

Employment. The unit’s administrative assistant receives a copy of the admission letter. He or she asks the future students for the usual required documents (CV, passport copy, etc.). Note that the applicant already provides these documents at the beginning of the process, in the registration phase. The documents must be provided again because there is limited exchange of information between ISA and SAP. These documents, together with the admission letter, are sent to the HR assistant, who is responsible for preparing the contract and arranging for the visa application, if needed. Once the contract is ready, it is sent for signature to the future PhD student and new records in the SAP human resource and finance management software modules are created.

4.3 The SEAM Workshop

The goal of the workshop organizer was to train the IT managers of the main applications on how to apply a service-oriented view to their application, using SEAM as a modeling method. A side goal was to make the participants aware of some of the technical and people issues concerning the PhD process and to prepare a follow-up workshop to address these issues (such as data integration between the registration and employment).

The workshop brought together six IT managers (e.g. SAP and ISA managers), the head of central IT and the person in charge of IT governance. The workshop was managed by one of the authors (Alain Wegmann) with the help of one of the co-authors (Gorica Tapandjieva). While writing this paper, we noticed that Alain Wegmann had three roles in this workshop: (1) workshop facilitator and SEAM trainer, (2) EPFL enterprise architect, (3) professor who hires PhD students. Ms. Tapandjieva had two roles: (1) SEAM trainer assistant, (2) Master’s student at EPFL and applicant for a PhD position at EPFL. She had, at the time of the workshop, a pending application in the PhD hiring process.

The workshop was held in the following way:

First, the participants expressed their expectations from the workshop. They were quite a few. For example, learning how to use SEAM to model services, finding ways how to work better with colleagues, or simply attending the workshop to see what comes out of it.

Next, we asked all participants to present the challenges they faced in managing their applications. The major challenges were: (1) understanding what the term “business” meant in business and IT alignment, (2) defining who are the relevant representatives of the about 10’000 EPFL users and (3) understanding what is the IT and business strategy of EPFL.

We then introduced the example of the PhD hiring process. We provided a two page textual description, a sequence diagram of the detailed process and a file with a copy of all documents from Ms. Tapandjieva’s application. We briefly introduced some of the SEAM principles: how to model systems, services, and processes. The participants worked in three groups (2 groups of 3 and one of 2 participants) and had to make a SEAM model of the PhD hiring process. We concluded with a debrief session and a sketch of a SEAM model made by Alain Wegmann. The goal was to encourage participants to practice SEAM (and thereby to understand the difficulties in using it) and then to show how a SEAM modeler would create a model that exposes the issues they had identified at the beginning of the workshop.

We ended the morning with a debrief session during which the participants said they liked the concreteness and the dynamic aspects of the method. Some participants found that the models were “more messy” than the ad-hoc ones they would normally make. Systemic models often appear less simple than add-hoc ones, who are frequently over-simplified.

In the afternoon we created a group-wide model of the PhD process. We discussed the technical and the organizational issues raised by a transition to a service approach. Figure 3 is a picture of the group-wide model that we created together.

The day ended up with a debrief session in which the participants agreed on the technical and organizational issues to address in moving to a service approach. Some raised the concern that we did not find a solution to these issues, but this was not planned for this workshop. It was also clear that a follow-up workshop should formally include more business users.

In the morning the IT managers made their model in three separate groups. They based their model on the sequence diagram of the detailed process we gave them. So they all analyzed the overall process (i.e. the three phases). One of the models happened to be quite similar to the group-wide model shown in Figure 3. The second model represented the point-to-point interactions in the process, a high-level view of the sequence diagram. It did not show the three phases identified in the group-wide model. Most notably, the model did not include the management of the admission letter, probably because this phase is not supported by an IT system. The third model represented the existing organizational boundaries within EPFL. The phases were represented within these boundaries. We recall that in SEAM we represent service systems, therefore these boundaries were not supposed to appear in this model.

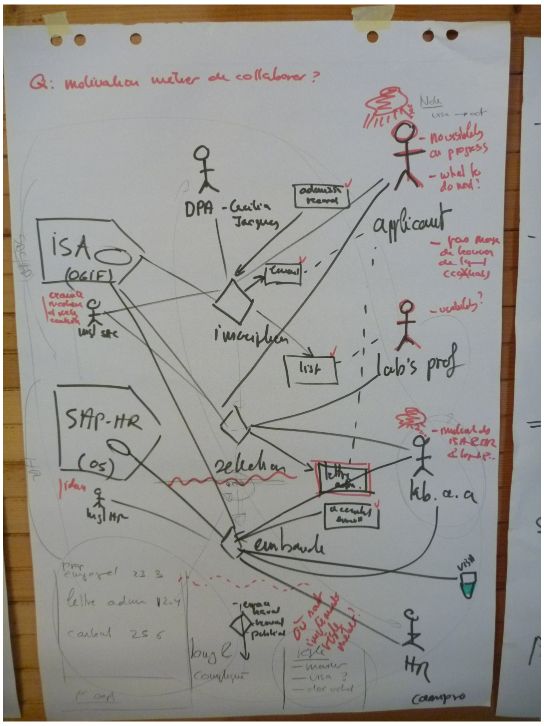

Figure 3: PhD Hiring SEAM model developed during the SEAM workshop.

4.4 The importance of the Systemic Modeling Paradigm for the Workshop

Federating different models and different conceptualizations: The three different models made by the three different groups were the result of three different stakeholders’ conceptualizations.

All the models were valid but seemed incompatible with one another. The systemic modeling paradigm helped us to not quarrel about who is right or wrong but to accept each model as a *bona fide* representation for the person or people who created it.

To design a common process, it is important that all stakeholders share the same model. This means that the reconciliation of their disparate conceptualizations is necessary. Changing people conceptualization, the way they see the world, is a known as a very difficult task. Axiology helps here for guiding the enterprise architect in this difficult task and in the choices that are inevitable in selecting what to represent in the common model.

If there is no conceptualization it will not be in the model: Each of the three groups modeled the overall process (with one group mostly focusing on the IT support). If we would not have given them the sequence diagram prior to the modeling exercise, it is very likely that the Selection phase would not have been represented because none of the IT managers provided support for this part of the process. None of the IT managers had "Selection" in their conceptualization.

The sequence diagram of the process was created by Ms. Tapandjieva who interviewed several stakeholders of the process and collected evidence about it before the workshop. In addition, Ms. Tapandjieva was also a PhD applicant and her application was somewhat "stuck" in the Selection phase for a few months. So she was able to testify on the importance of this part of the process for an applicant. Thanks to the testimony of Ms. Tapandjieva and to the collected evidences, it was possible to model the "Selection" phase and to identify the related issues (e.g. there is no specialized IT support, applications could get stuck in this phase, etc.).

Each IT manager could model with precision the phase supported by the application he was responsible for. This phase relates directly to his conceptualization because it corresponds to his specialization. One of the challenges during the workshop was to enable all IT managers to represent their phase at the same level of detail as the other phases.

One of the participants offered an additional conceptualization. His training as an auditor enabled him to discover a flaw in the sequence diagram of the process by attentively analyzing the dates of the documents provided as evidence. Without this specialization the sequence diagram would have not been challenged.

In summary, to have the viewpoints of the multiple stakeholders (including the non-IT ones) it is essential to understand the issues related to the process. This includes the IT issues. For example, the applicant has to submit his documents to ISA and SAP. This leads to errors and delays. A technical solution can be found by linking ISA and SAP. This problem can be identified only if the process is analyzed end-to-end. So, all viewpoints are necessary.

The use of concrete *evidences:* Some of the documents collected by the way the process is executed leads to major issues for the applicant. For example, the applicant does not receive the necessary documents on time to find housing. This level of concreteness motivates the other stakeholders to address the issues. They can relate to the applicant's problems. All the participants were able to relate to the feeling the applicant has when the document that would allow him or her to find an apartment is not received on time. This is much more concrete than the concept of "hard to find an apartment" that would usually be found in abstract models.

Once more, without the evidence provided by the documents collected by Ms. Tapandjieva the auditor would not have found the flaw in the sequence diagram.

5 CONCLUSIONS

In this paper we emphasized the need to have a philosophical grounding for business-IT alignment because it is a crosscutting concern that potentially requires the collaboration of the entire organization. We described one such grounding, called the systemic modeling paradigm, which is based on general systems principles, and is the foundation of SEAM, an enterprise architecture method. The main originality of the systemic modeling paradigm is its breadth. It proposes 4 dimensions for underpinning a general-purpose method that can be effectively used in concrete projects. These dimensions are, theory, philosophy, methodology and discipline specific theories. Together they enable to transcend the divisions within an organization, while also understanding the specificities of each department or individual stakeholder. It is our hope that other researchers would use this paradigm or propose different paradigms to provide a philosophical foundation for their methods, an aspect that business and IT alignment urgently needs.

REFERENCES

Banathy, B. H. and Jenlink, P. M. (2004). Systems inquiry and its application in education., In Jonassen. D.H. (Eds.), Handbook of research on educational communications and technology, (pp. 37-57). Lawrence-Erlbaum,

Beyer, H. and Holtzblatt, K. (1998). Contextual Design: Defining Customer-Centered Systems. San Francisco, CA: Morgan Kaufmann.

Chan, Y.E. and Reich, B.H. (2007). IT alignment: What have we learned? Information Technology, 22, 297-315.

Checkland, P., Holwell, S. (1998). Information, Systems and Information Systems, making sense of the field. Chichester, UK: Wiley.

Gruber, T. (2009). Ontology. In Liu, L. and Özsu, M.T. (Eds.), Encyclopedia of Database Systems. Springer-Verlag.

Lankhorst, M.M., Proper, H.A. and Jonkers, H. (2009). The Architecture of the ArchiMate Language. Proc. BPMDS 2009 and EMMSAD 2009, LNBIP 29, Springer.

Lemos, N.M. (1999). Value theory In Audi, R. (ed.), The Cambridge Dictionary of Philosophy. Second Edition, Cambridge University Press, Cambridge UK.

Luftman, J., McLean, E.R. (2004). Key Issues for IT Executives, MIS Quarterly Executive, 3(2).

Merriam-Webster, (2013). http://www.merriam-webster.com/dictionary/paradigm, accessed May 2013.

Mintzberg, H., Ahlstrand, B. and Lampel J. (1998). Strategy Safary – The complete guide through the wilds of strategic management. Prentice Hall.

Regev, G., Hayard, O. and Wegmann, A. (2011). Service Systems and Value Modeling from an Appreciative System Perspective. In *IESS1.1, Second International Conference on Exploring Services Sciences*, Springer.

The Open Group. (2009). The Open Group Architecture Framework (TOGAF) Version 9, The Open Group.

Vargo, S.L., Maglio, P.P. and Akaka, M.A. (2008). On value and value co-creation: A service systems and service logic perspective. European Management Journal 26(3), 145–152.

Vickers, Sir G. (1968). Value Systems and Social Process. Tavistock, London

von Bertalanffy, L. (1968). General System Theory. New York: George Braziller.

Wegmann., A. (2003). On the Systemic Enterprise Architecture Methodology (SEAM). International Conference on Enterprise Information Systems.

Weinberg, G. M. (1975). *An Introduction to General Systems Thinking*, Dorset House.

Zachman, J.A. (1987). A framework for information systems architecture. IBM Systems Journal 26(3).