

Requirements Modeling in SEAM: the Example of a Car Crash Management System

Alain Wegmann*, Biljana Bajić-Bizumić*, Arash Golnam*, George Popescu*
Gorica Tapandjieva*, Anshuman B. Saxena*, Maedeh Yassae†, Gil Regev*
* École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
†Université de Lausanne (UNIL), CH-1015 Lausanne, Switzerland

Abstract—This paper presents how business and IT requirements are captured with the Systemic Enterprise Architecture Methodology (SEAM). The method is applied to the Car Crash Management System (CMS) - Software Product Line (SPL) case study. The existing business situation is analyzed. We identify the problems and list relevant solutions. We select one of these solutions for which we define the business and the IT requirements. We then present the two components of SEAM used in this paper, goal-belief and behavior modeling. We end the paper by presenting the systemic foundations of SEAM.

I. INTRODUCTION

The Systemic Enterprise Architecture Method (SEAM) is a modeling technique developed at EPFL. The first paper about SEAM was published in 2003 [1]. SEAM is used in consulting for business-IT alignment [2], [3], enterprise architecture [4] and business strategy [5]. It is used for teaching enterprise architecture, service-oriented architecture, requirements engineering [6], [7] and business strategy for IT services.

The theoretical foundations of SEAM are in General Systems Thinking (GST) [8] and in RM-ODP (ISO Standard 1995) [9]. GST seeks to define general principles that can be applied to any phenomena across established disciplines. RM-ODP is a software engineering ISO standard that provides the underlying definitions for the SEAM concepts (e.g. object, behavior, state, property) [9]. SEAM is rigorously defined based on these systemic and software engineering concepts and federates multiple modeling techniques (such as discrete behavior and goals). Thus, SEAM provides a consistent set of modeling principles and constructs to model an enterprise at different abstraction levels both from organizational (company, department, IT infrastructure, etc.) and behavioral standpoints (services and processes at different granularity) [1].

SEAM is inspired by the Catalysis approach [10], the Larman/Fusion process [25] and by the Unified Modeling Language (UML) [11]. The Catalysis approach is a component-oriented development method based on the two main concepts: object and action. Some of the main principles overtaken from Catalysis and adapted in SEAM are: hierarchical model structure, functional and organizational refinement, localized and distributed action [10]. SEAM proposes a concrete implementation of Catalysis and extends Catalysis to business modeling. Moreover, SEAM notation builds on the UML notation and proposes one kind of diagram that includes a subset of the element kinds found in the 14 UML diagrams (UML v. 2.4.1). A SEAM diagram augments the UML deployment diagram, use case diagram and class diagram.

SEAM is used in early design phases for early requirements gathering. It is a method applied for system pre-design and can complement other requirements engineering approaches that focus more on requirement specification and implementation, such as the UML and the Business Process Modeling Notation (BPMN). Its purpose is not to show the low-level design, but to delimit the problem, analyze and discuss the stakeholder's viewpoints, show their objectives and the processes in which they are involved, without showing all the details. In practice, the method is used with pen, paper, post-its, flip-charts, etc.

Based on our research on the application of SEAM in enterprise architecture and requirements engineering, the industry practitioners found the graphical representations and the notational elements in SEAM to be cognitively effective. In addition, our experience shows that the systemic principles underlying SEAM, such as black-box and white-box representations of systems and their behavior, provide useful insights into modeling an organization and building a common understanding of the problem being analyzed.

SEAM provides two representations that we use to model both the business and IT context:

- **Goal-Belief model** - Describing the motivation of the business and of the IT stakeholders. We model the beliefs of actors, and show how the reality of the situation and the general disposition of actors drive them to execute actions and achieve common goals.
- **Behavior model** - Describing the services of the business and IT stakeholders, and their corresponding implementation processes. We focus on key stakeholders and the main services delivered to the customer.

The typical analysis and design made using SEAM contains the following steps:

- *Step 1:* Systems identification using the Behavior “as-is” model. Here we model a nested hierarchy of systems i.e. the systems that compose an organization and the systems in which the organization is embedded.
- *Step 2:* Problem identification with the Goal-Belief “as-is” model. We model the goals and the beliefs of the key actors to identify the potential problems.
- *Step 3:* Analyzing possible solutions and choosing one. We scan the solution space and choose the solution that addresses the problem identified in the previous step. Here we take into account constraints, such as human and/or IT resources, time, etc.

- *Step 4: Solution description with the Behavior “to-be” model.* In this step we develop the Behavior “to-be” model to represent the solution. The changes to the “as-is” model are shown by means of signs and color coding. This solution representation can be simulated, prototyped and possibly mapped to BPMN for obtaining an executable application.

The paper is organized as follows: In Section 2, we illustrate the steps of the SEAM analysis and design process. We provide brief examples extracted from our complete models of the Crash Management System (CMS) case study. In Section 3, we present the SEAM theory underlying the SEAM models. In Section 4, we present our conclusions.

II. CAR CRASH MANAGEMENT SYSTEM CASE STUDY

In this section, we apply the SEAM design steps to model the requirements in the CMS case study.

A. Step 1: System Identification with the Behavior “as-is” Model

The first step of the SEAM approach is to identify the service systems that constitute the universe of discourse. We identify the hierarchy of the systems for the CMS case study, which compose an organization (company, departments, management system, IT infrastructure, etc.) and in which the organization is embedded (market segments, etc.). The systems are identified based on the services and processes that are modeled. Figure 1 illustrates the hierarchy of the systems in the CMS case study¹. This model corresponds to the current (“as-is”) situation thus allowing us to identify the main problem. The following organizational levels are modeled in SEAM:

- **Public safety segment** - representing the main stakeholders providing the crisis management service;
- **Police Station (PS) and Fire Station (FS) organizations** - defining the internal organization and processes of the organizations providing the resources for rescuing;
- **Police Station and Fire Station local ITs** - showing the IT infrastructure, databases, protocols and applications of the organizations that rescue the victim.

The goal of system identification is two-fold: first, to delimit the problem and identify the stakeholders relevant for the part of the reality that we model and second, to organize the systems and define the correct boundaries corresponding to their service offerings and process implementation. Therefore, we show the system organization in the behavior diagram, which represents the services and process of the business and IT stakeholders in which the systems are involved.

In the first organizational level, we show the “Public Safety Segment” (PSS) (Figure 1a), including all the stakeholders participating in the crisis management service. The Police and Fire Station are represented as separate systems PS and FS, because in the current situation they offer the intervention

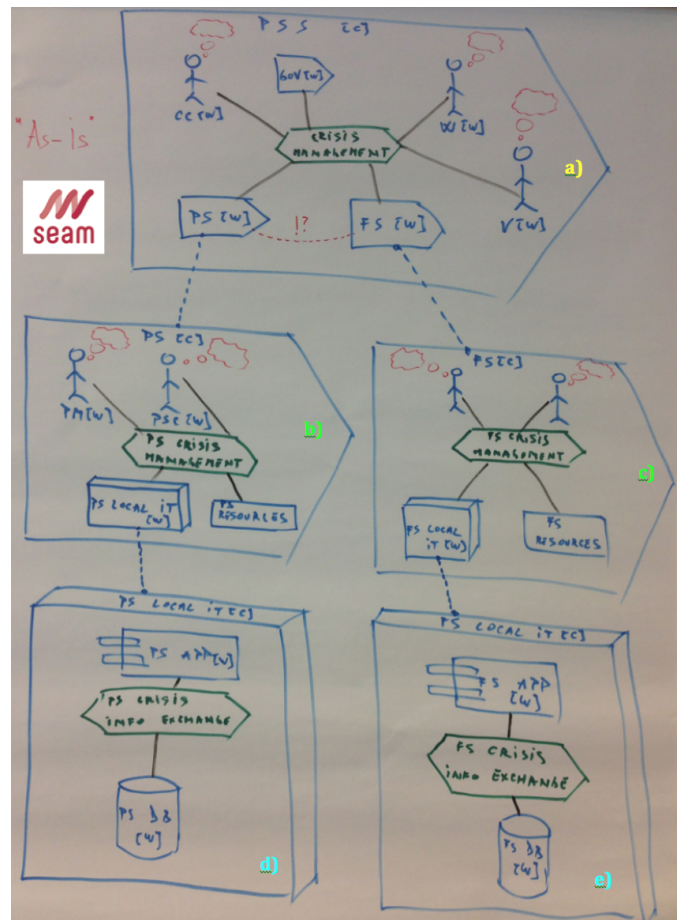


Fig. 1: Behavior “as-is” model done with markers on a flipchart

services separately, i.e. there is no single point of contact in case the accident happens.

In the second organization level, we reveal the internal organization of the PS and FS (Figure 1b,c), showing e.g. Policeman, Police Station Coordinator, Police Station IT system (without revealing the details of it) and resources they need for an intervention. We also show the services and processes in which they are involved (Figure 1b,c).

In the last organization level, we show the details of the local IT systems of the PS and FS (Figure 1d,e), including their databases, software components, etc.

With these three organizational levels, we have a full comprehension of the situation we model. They enable designers to discuss the current situation and possible solutions from three different perspectives, spanning from the business organization of the public safety segment down to the technical details of the local ITs: the FS and PS local IT systems can be supposed to exist or be later implemented.

B. Step 2: Problem Identification with the Goal-Belief “as-is” Model

Once we have identified the systems in the Behavior “as-is” model, we explain and analyze the Goal-Belief “as-is” model of the “Public Safety Segment”.

¹Figure 1 reflects how SEAM models are drawn in workshops with industry participants.

In Figure 2, we show the Goal-Belief of John, the witness, including his main goal (at the left), reality judgments, value judgments (at the top) and the resulting action judgments (at the bottom). The main goal shows the main objective of the system. It is often a maintenance goal that shows how the system maintains its identity, i.e. survive in a changing world. The maintenance goal represents a condition that remains constant for the system, i.e. keeps its identity [12]. In case of the witness, the condition that maintains his identity is to “be present at the crisis scene”. This means, we identify the witness as someone who is present at the crisis scene.

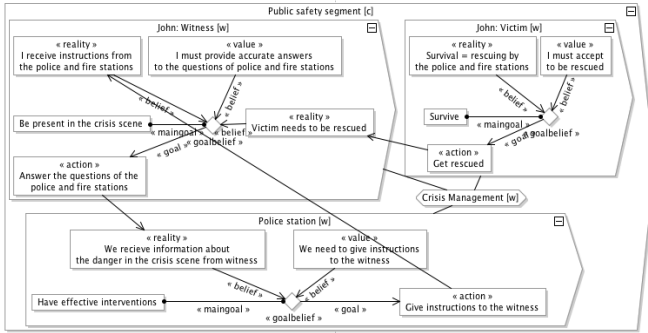


Fig. 2: Goal-Belief “as-is” model for the witness

The reality and value judgments correspond to the beliefs of the witness. Based on these beliefs and his main goal, John decides to react to the changes in the environment by doing the actions shown in Figure 2. A reality judgment describes what the witness is ready to notice in himself and his environment. A value judgment describes how what he noticed will affect present and future relations within the organization and its environment. We see a value judgment as a “trigger” for an action judgment which describes what behavior is appropriate based on what has been noticed.

For example, John’s observation of the environment is shown in the reality judgment “I receive instructions from the PS and FS”. Based on his value judgment that “I must provide accurate answers to the questions from the PS and FS”, he decides to act: “Answer the questions from the PS and FS”. We use a line with an arrow connected to the mentioned reality judgment to represent the fact that this reality judgment is coming from the witness’s environment, in this case PS and FS, giving John the instructions.

Notice that the system for the witness is called “John: Witness”, showing another pattern in SEAM: the names of real people and organizations are mentioned. In this case, we modeled the witness and victim as two separate roles that can correspond to the same person, e.g. John. A victim is seen as someone whose main goal is to survive, and the main action judgment is to be rescued while the witness is someone present at the crisis scene and providing the information to the PS and FS. One person can correspond to two or more roles at the same time. For example, John as a victim might be someone injured who needs to survive and be rescued. At the same time, being conscious, he is a witness and can alert the PS and FS and provide useful information to them. However, if he is unconscious he can only be the victim. In this way, we distinguish the two clear roles in the system that can be

dedicated to the same or different persons.

Similarly to this, we model the goal-belief for all the other participants, showing their internal view of the crisis management. Therefore, for each of the systems we use their specific internal vocabulary. For example, the action judgment “Establish policies for coordinating PS and FS” in Government Agency, becomes a reality judgment in the PS “We receive our rules of engagement in case of danger from the government”, showing that the government policies correspond to the rules of engagement for the PS.

Once we have modeled the internal perspective of each system on the current situation, we analyze the connections between different systems and find the problem. The victim needs to be rescued and the witness notices and provides information about the crisis to the PS and FS. The Government Agencies establish the policies to the PS and FS. Based on these policies and the information about the crisis they communicate to each other, organize the number of vehicles and plan for the route. The communication compromiser that we model as a journalist accesses and shares the information about the crisis. This information reaches the Government through him, which concludes that the PS and FS are not well coordinated and complains to them. Because of this, we show an action judgment in the PS that they intervene ineffectively due to bad coordination, which is in contradiction with their main goal: effective intervention.

We conclude that the main problem is the coordination of the PS and FS. Therefore, the interventions are ineffective. If we also analyze the technical organizational levels of the Behavior “as-is” model, we determine the possible technical details causing the problem, such as data inconsistency between the local databases of the FS and PS. All this can cause the communication between the victim, witness, PS and FS to become too complex and unmanageable. Therefore, more time might be needed for the victim to be rescued.

C. Step 3: Analysis of Possible Solutions and Choosing One

Analyzing the Goal-Belief “as-is” and the Behavior “as-is” models containing the three organization levels, allows us to draw possible solutions from the business down to technical perspective. The process would be more effective if there is only one organization providing the common service for the crisis management. The possible solutions to achieve this are:

Solution 1 - Senior Engagement Officer (SEO): From a pure business perspective, the solution could be to introduce a new person, e.g. SEO, who would rapidly go on-site and coordinate the PS and FS.

Solution 2 - Android/iPhone application: From the point of view of the organization providing the resources for the rescue, there could be a mobile (Android/iOS) application. This application can help the witness take pictures that can be sent to the FS and PS ². These pictures can help the PS and FS to react quickly.

Solution 3 - bCMS: Finally, from the technical perspective, we could introduce the middleware that could facilitate data synchronization between the local databases of the PS and FS.

²Touring Club Switzerland (TCS) provides a similar solution.

As it can be seen, the SEAM model provides a way to communicate and discuss the alternative solutions and gain a common understanding of the team working on the project. In this case, we choose the solution 3 (*bCMS*) and the synchronization of the local databases of the PS and FS.

D. Step 4: Solution Description with the Behavior “to-be” Model

In the Behavior “to-be” model we show the main solution by designing one organization (PS and FS in Figure 3a) that includes both the FS (Figure 1c) and PS (Figure 1b) together participating to provide a common service for the crisis management. This does not mean that both organizations are merged, but that they provide an integrated service together. In the next layer, we show the internal details of this organization (Figure 3b), including the Policeman, Fireman, and the other roles, providing all together the service to the victim. On the IT level, we have the local IT systems of both PS and FS (Figure 3c, d) and the additional system in charge for synchronization of the two local IT systems (Figure 3e). In this way, the witness has only one access point for communication when the accident happens, which makes the communication more effective. The communication between the PS and FS is hidden inside the system rescuing the victim.

We see the solution of the problem in the implementation of a *bCMS* system that will rely on the IT infrastructure of the two key actors: the PS and FS departments. Since the *bCMS* system is distributed, it will rely on the data provided by the two systems. Thus, *bCMS* will facilitate only the communication of such critical information. From a diagrammatic representation, we use a “+” sign on an object to represent what is added by the solution, and a “~” sign for what needs to be changed.

In the SEAM model, we do not show the details of the processes and services, as the possible (full) implementation is not the purpose of the method. It is possible to do it, but it is rarely used at this stage of a project. We focus on the main stakeholders, responsibilities, problems and solutions.

To support an implementation, we can map the processes identified in SEAM into BPMN. In order to facilitate agile service design SEAM also provides ways to prototype and simulate models: it enables business experts to validate the defined services and models through an animator tool. This allows business experts to “execute” the model as if it was already implemented. Also, it allows the designer to find the design mistakes in the early design phase. We provide the formalization and simulation of both the Goal-Belief and the Behavior model [13] in the Alloy language [14]. In addition to the Alloy simulation, we provide also the prototyping of the behavior models into the given target language, e.g., Java [15]. In addition to the previous contribution, this allows business experts to review the prototype of the application corresponding to the behavior model in SEAM.

III. SEAM MODELING APPROACH

SEAM has an explicit systemic paradigm, as explained in [1], which sets it apart from most other enterprise architecture methods. This systems paradigm [16] has the following components: systems theory, systems philosophy and systems methodology. Zachman [17] is an exception as it provides

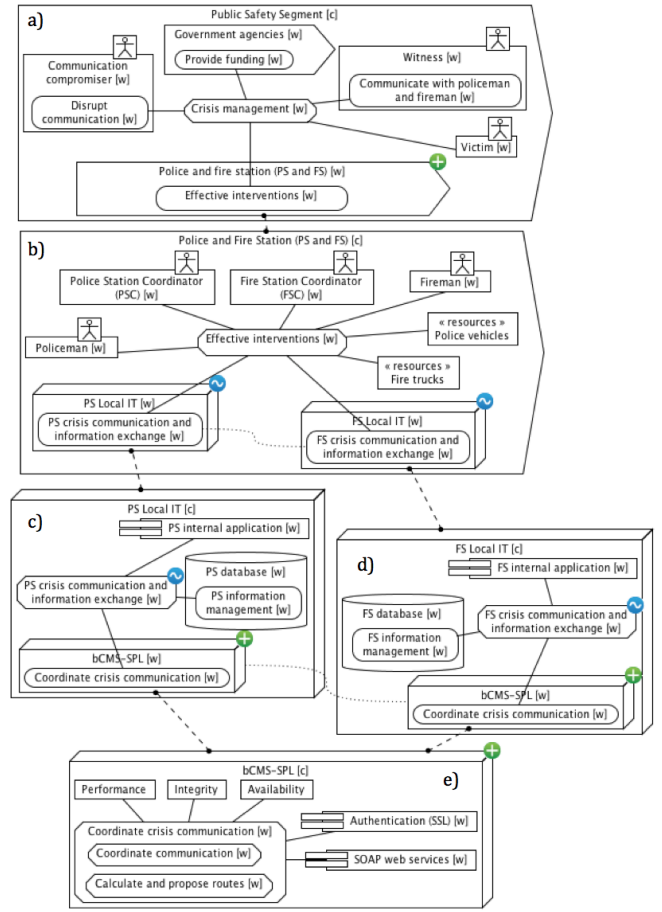


Fig. 3: The Behavior “to-be” model of the “Public Safety Segment” represented at the same levels as the “as-is” showing the needed changes to implement the proposed solution

some epistemological considerations in the form of a building architecture metaphor. Soft Systems Methodology (SSM), which is built on a systemic paradigm, is another notable exception [18].

A. Systems Philosophy

Systems philosophy as described by Bánáthy and Jenlink [16] is composed of ontology, epistemology and axiology. In SEAM, epistemology describes the relationship between the model and reality. For example, is the reality perceived or objective? Ontology describes the model elements. Thus, it offers the method’s vocabulary. Examples of ontology terms are: organization, service, process, etc. Axiology is concerned with the choices made by the modelers when they select some aspects of reality (i.e. the systems and the level of abstraction) to be modeled.

The constructivism principle, on which the SEAM epistemology is based, explains how people construct models and why they perceive a hierarchical reality. In our example, the project starts by representing the overall car crash management situation as a hierarchy of systems with relationships between them. The system entities perceived in the reality are represented in the model in different organizational levels. This

crucial step is necessary to build a common, relatively simple, understanding of the situation from the ad-hoc representation of reality. Once the systems are defined, the services and processes can be macroscopically represented. We do not represent the detail of the service or the process, but only their overall result.

A SEAM model represents systems that people perceive in their reality. These are represented as modeling elements that we call working objects. We can represent the externally visible behavior or the motivation of the systems. For this we represent the working object as a whole (black box). The externally visible behavior is represented as a working object's service (e.g. PS and FS crisis management in Figure 3a). The motivation is represented as the working object's goals and beliefs (e.g. Figure 2). We can also represent the systems construction (i.e. the component working objects). In this case, we show the working object as a composite (or white box, transparent box). We can then model the process that implements a service (e.g. PS and FS crisis management in Figure 3b). We can further analyze the goals and beliefs of the component working objects (not done in the case study).

The working objects in our models appear with the appropriate business term, e.g., markets, segments, value networks (group of companies), companies, departments, people, IT system, IT infrastructure, etc. We can represent any kind of systems with a working object. The same modeling rules apply regardless of the nature of the modeled system. For example, the service provided by a company is described with the same modeling technique as the service provided by its IT infrastructure.

As is customary in enterprise architecture methods, we model the situation as-is, and then the situation to-be. This is useful for analyzing the problems (in the as-is) and for describing the solution (in the to-be). However, the system structure might change between the as-is and the to-be. As explained above, systems are conceptualized to represent the reality (actually a simplified view of the reality). When we change our understanding of the reality, the perceived systems change as well. So a to-be model might have different systems compared to an as-is model. This is one of the important specificities of SEAM. We detail next the theoretical foundations on how we model behavior and goal-belief. We end with a description of SEAM diagrammatic principles.

B. Behavior Ontology

A working object as a whole provides a service. A service is an action defined with pre and post-conditions. These pre and post-conditions modify the state of properties (not shown in the example). A working object as a composite implements a process that combines the services offered by the component working objects as wholes. For example, the "Police and Fire Station" working object as whole (Figure 3a) provides the service "PS and FS crisis management". This service is implemented by the process with the same name in the "Police and Fire Station" as a composite (Figure 3b). Inside the Police and Fire station as a composite, the Police Station Coordinator, the Fire Station Coordinator and six other working objects (all seen as whole) participate in the process (Figure 3b).

The service and the process are related by a refinement relationship: the process is a refinement of the service. Refinement relationships are defined in Refinement Theory [19]. A formal semantics is defined for the behavior diagram [19] and is simulated using the Alloy specification tool [20]. The service terminology comes from the best practices defined in the Information Technology Infrastructure Library (ITIL) [21].

C. Goal-Belief Ontology

The Goal-Belief model [22] makes explicit the actors' motivations (Figure 2). The actors' beliefs are represented as SEAM properties and based on Vickers Appreciative System [23] model. Vickers created the Appreciative System as a way to represent human and organizational behavior [24]. Based on this model we define three kinds of related beliefs:

- **Reality judgment:** describes what a system is ready to notice in itself or in its environment. Reality is a kind of belief. There is no equivalent in other goal-oriented RE methods. For example, in Figure 2, we see that the witness has a readiness to receive instructions from the police and fire stations.
- **Value judgment:** describes how what is noticed will affect the system behavior. It is related to norms. If a reality is outside an expected norm, the system will react. Value is another kind of belief. There is no equivalent in other goal-oriented RE methods. In Figure 2, we see that the witness understands that he must provide accurate answers, which commits him to action.
- **Action judgment:** describes the resulting behavior. It is a consequence of a set of reality and value judgments. An action corresponds to a goal in other goal-oriented RE methods. In our example, the witness answers the questions of the police. This is logically related to his reality and value judgments.

The Goal-Belief and the Behavior models are complementary as they offer different insights into the problem domain. Also, the Goal-Belief has a formal semantics defined in Alloy.

D. Principles on Graphical Visualization

We have defined a set of principles on how to represent a SEAM diagram. Below are a few of them.

Explicit context: The context is always made explicit in a SEAM diagram. For example, in Figure 3, we see the contexts in which the systems exist. Systems are positioned inside a boundary, except the top one which gives the overall context. For example, "Public Safety Segment [c]" is the top context. Inside "Public Safety Segment [c]", there is "Police and Fire Station [w]", which corresponds to "Police and Fire station [c]". Inside "Police and Fire Station [c]", there is "Police Local IT [w]", and so on. In many notations, such as UML, the context is implicit. For example, in the UML use case diagram [10], it is possible to hide the IT system boundary [25]. The underlying principle that explains why UML allows hiding the IT system boundary is called the Occam-razor principle. This principle express that a succinct theory is better than a verbose one. The boundary is not considered as an important concept,

so it can be hidden. In a systemic model, the boundary is one of the most important concepts, so it has to be visible.

Explicit hierarchy: The SEAM organizational hierarchy captures systems' construction. The functional hierarchy captures behaviors' structure (services or processes). We represent hierarchies in SEAM as boxes inside boxes. We do not have, as in UML, composition relationships. With this we put an emphasis on which concepts are hierarchical and which ones are not. The hierarchy makes also explicit the context in which the element is defined.

Explicit roles: The key actors are represented in SEAM diagrams with their explicit roles. It is possible to have the "same" actor with different roles (in different systems). In our example, the witness and the victim can be (or cannot be) the same person. This is extremely useful to analyze conflicts of interests and resource issues.

Explicit mapping to reality: In SEAM diagrams, we show concrete projects, people, services, departments, etc. If possible, we use the picture of the real people, companies, products, etc. This helps make the model concrete. If we make a business model, we analyze one representative customer with a name.

IV. CONCLUSIONS

The originality of SEAM is to consider that both the business systems and the IT systems need to be analyzed and designed in parallel. Other methods usually consider business analysis as a prerequisite for IT design. These methods do not include any business design. For such methods, the IT serves the business. In SEAM, the co-evolution of business and IT serves the company.

SEAM is designed to model complex organizations in which there are conflicting viewpoints. The construction of the initial behavior "as-is" is the technique to get all stakeholders agree on existing systems. This becomes a common ground for the project, for which we use SEAM to construct a goal-oriented "as-is" and then a "to-be" architecture for CMS. We describe how SEAM is applied to CMS through a series of steps including the selection of a design solution.

We mostly use SEAM in early requirements phase, to scope the project and to create a consensus between the business and the IT stakeholders on the issues to address, the possible solutions and the selected one. IT development and implementation phases are based on the SEAM early analysis and design specifications. SEAM is used for consulting and teaching. As we analyze and design business and IT with a similar set of concepts (systems, behavior and goal-belief, models), it is possible to teach, in a one-semester course (3 or 6 periods per week), the analysis and design of the business, organization and IT levels [7].

REFERENCES

- [1] A. Wegmann, "On the Systemic Enterprise Architecture Methodology (SEAM)," in *ICEIS*, 2003.
- [2] A. Wegmann, P. Balabko, L.-S. Lê, G. Regev, and I. Rychkova, "A method and tool for business-it alignment in enterprise architecture," in *Proceedings of the CAiSE*, vol. 5. Citeseer, 2005, pp. 113–118.
- [3] A. Wegmann, P. Julia, G. Regev, O. Perroud, and I. Rychkova, "Early requirements and business-it alignment with seam for business," in *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*. IEEE, 2007, pp. 111–114.

- [4] A. Wegmann, G. Regev, I. Rychkova, L.-S. Lê, J. D. De La Cruz, and P. Julia, "Business and it alignment with seam for enterprise architecture," in *Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International*. IEEE, 2007, pp. 111–111.
- [5] A. Golnam, G. Regev, J. Ramboz, P. Laprade, and A. Wegmann, "Systemic service design: Aligning value and implementation," in *Exploring Services Science*. Springer, 2010, pp. 150–164.
- [6] A. Wegmann, G. Regev, J. D. De la Cruz, L.-S. Lê, and I. Rychkova, "Teaching enterprise and service-oriented architecture in practice," in *Proceedings of the Workshop on Trends in Enterprise Architecture Research (TEAR 2007)*, 2007.
- [7] G. Regev, D. C. Gause, and A. Wegmann, "Experiential learning approach for requirements engineering education," *Requirements engineering*, vol. 14, no. 4, pp. 269–287, 2009.
- [8] G. M. Weinberg and J. Wiley, *An Introduction to General Systems Thinking*. Wiley New York, 1975.
- [9] A. Wegmann, L.-S. Lê, G. Regev, and B. Wood, "Enterprise modeling using the foundation concepts of the rm-odp iso/itu standard," *Information systems and e-business management*, pp. 397–413, 2007.
- [10] D. D'Souza and A. Wills, *Objects, components, and frameworks with UML - The Catalysis approach*, 4th ed. Addison-Wesley, 2001.
- [11] M. Fowler, *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional, 2004.
- [12] A. Wegmann and G. Regev, "Where do goals come from: the underlying principles of goal-oriented requirements engineering," in *Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on*. IEEE, 2005, pp. 353–362.
- [13] A. Wegmann, L.-S. Lê, I. Rychkova, and G. Regev, "An Example of a Hierarchical System Model Using SEAM and its Formalization in Alloy," in *EDOC Conference Workshop, 2007. EDOC '07. Eleventh International IEEE*, oct. 2007, pp. 260–268.
- [14] D. Jackson, I. Schechter, and I. Shlyakhter, "ALCOA: The Alloy constraint analyzer," in *Proceedings of the 22nd International Conference on Software Engineering (ICSE)*, Limerick, Ireland, June 2000.
- [15] B. Bajic-Bizumic, C. Petitpierre, H. Chi Huynh, and A. Wegmann, "A Model-Driven Environment for Service Design, Simulation and Prototyping," pp. 200–214, 2013.
- [16] B. H. Banathy and P. M. Jenlink, "Systems inquiry and its application in education," *Association for Educational Communications and Technology*, pp. 37–58, 2004.
- [17] J. A. Zachman, "A framework for information systems architecture," *IBM systems journal*, vol. 26, no. 3, pp. 276–292, 1987.
- [18] P. Checkland and J. Scholes, *Soft Systems Methodology in Action*. Wiley, 2001.
- [19] I. Rychkova, G. Regev, and A. Wegmann, "Declarative specification and alignment verification of services in itil," in *Enterprise Distributed Object Computing Conference Workshops, 2008 12th*. IEEE, 2008, pp. 127–134.
- [20] D. Jackson, *Software Abstractions- Logic, Language and Analysis*. MIT Press, 2011.
- [21] "Information Technology Infrastructure Library," <http://www.itil-officialsite.com/>.
- [22] G. Regev and A. Wegmann, "Defining Early IT System Requirements with Regulation Principles: the Lightswitch Approach," in *Requirements Engineering Conference, 2004. Proceedings. 12th IEEE International*. IEEE, 2004, pp. 144–153.
- [23] G. Regev, O. Hayard, and A. Wegmann, "Service systems and value modeling from an appreciative system perspective," in *Exploring Services Science*. Springer, 2011, pp. 146–157.
- [24] S. G. Vickers, *Value Systems and Social Process*. London, 1968.
- [25] C. Larman, *Applying UML and patterns*. Prentice Hall PTR, 2005.