Patterns for Value-Added Services Illustrated with SEAM

Gorica Tapandjieva, Aarthi Gopal, Maude Grossan, Alain Wegmann
École Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Switzerland
{gorica.tapandjieva, aarthi.gopal, maude.grossan, alain.wegmann}@epfl.ch

***Abstract*—The basic idea behind the provision of a service is to hide the service implementation details from the client in such a way that more value is provided to the client. Very often support services allow too much of their implementation details to be visible to customers resulting in poor value proposition. In this paper we describe a project that aimed towards the requisite abstraction of the support service implementation for the research funding at the Swiss Federal Institute of Technology (EPFL). We show the current service, the problem it creates for its customers, and propose four patterns for improving the value that customers receive from a requisite service abstraction. The four patterns are: provide a simple service interface, incorporate needed external actors into the service view of the organization, recognize new customers and use data and process linking technology. These patterns are applied in a to-be model.**

***Keywords*—service, service modeling, process modeling, business and IT alignment, SEAM, design patterns**

I. INTRODUCTION

Maglio et al. [1] define the concept of service in service science as “the application of resources for the benefit of another”. The IT Infrastructure Library (ITIL) [2], [3], provides a more concrete definition of a service as “a means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks”.

In our context ”facilitating outcomes customers want to achieve” means that a service provider must hide the details related to the service implementation used for obtaining the results that customers expect. If obtaining the same results using the service is as complicated for customers as not using it, then the service does not provide as much value as it could. Conversely, hiding all the details from the customer may not be feasible for technical or regulatory reasons. Also, some customers may not want that all the details be hidden from them. It is therefore necessary to define the right details to be hidden given the context in which the service is provided.

In this paper we do not discuss all these possibilities, we show only the right details corresponding to customers’ motivations. For this purpose, we propose four patterns that aid in the abstraction of the service implementation details:

- provide a simple service interface,
- incorporate needed external actors into the service view of the organization,
- recognize new customers, and
- use data and process linking technology.

We report on a case study from a project conducted at the Swiss Federal Institute of Technology (EPFL). EPFL is a university with an IT strategy focused on implementing services based on EA and architectural principles [4]. The case study is the application process for research funding, where we notice and apply our value-added service design patterns.

For the modeling of the case study, we use the Systemic Enterprise Architecture Methodology (SEAM) [5]. SEAM is a family of methods for strategic thinking, business/IT alignment and requirements engineering. We use SEAM to model service offerings, and service implementations (i.e. processes), while capturing values and motivations. We also show the difference between a service offering and a service implementation.

The paper is structured as follows. In Section II, we give the scope and description of the case study. In Section III, Subsection III-A and Subsection III-B, we show how we develop a SEAM as-is version of the case study and discuss it. We describe our perception of the stakeholders’ value and motivations in Subsection III-C. The proposed patterns are listed in Subsection III-D and applied on the to-be service model described in Subsection III-E. This to-be model shows the requisite abstraction of the service implementation. We list the related work in Section IV and, we finish with conclusions and future work in Section V.

II. CASE STUDY: APPLICATION PROCESS FOR RESEARCH FUNDING

The Swiss Federal Institute of Technology (EPFL) is one of the top international science and technology universities [6]. The research at EPFL spans many fields. An inevitable aspect of doing research is the need for funding. Here we look at a simplified, yet realistic application process for obtaining funds for a research project.

A. Process Description

The following actors, including people, organizations and IT applications take part in the application process for research funding:

- *Funding Organization (FO)* is an external source of funding. It can be a foundation, government department, corporation, company, private donor, etc. The reasons for funding vary from the desire for advancement in knowledge to obtaining profit. In both cases, the organization maintains close contact with its partners, because its image is very important.

As different organizations have different regulations for granting funds, researchers have to adapt the administration of their project (non-scientific part) to comply with the requirements of the FO. The FO makes research project acceptance decisions based on an internal competitive process that is out of the scope of this case study.

- *Researcher (Tom)* is a person at EPFL who is passionate about progress and wants to respond to society's concerns. With his laboratory, or together with partner laboratories, he has the competencies and ideas to carry out a research project. He applies for funds for his research project. These funds have certain policies and regulations to which the research project must comply.
- *Research Office (ReO)* is an organization within EPFL that promotes the quality and image of research at EPFL. For this purpose, the ReO provides to researchers all grant-related information, assists them with finances and offers them administrative support for their application process for research funding.
- *Technology Transfer Office (TTO)* is an EPFL organization that manages the intellectual property resulting from research. TTO also evaluates new inventions, negotiates and approves research contracts with industrial partners; it also ensures EPFL's rights and the patents that emerge from the research projects.
- *Dean of a School at EPFL (Jim)* is person who has significant authority over the activities that take place in the School. At EPFL there are seven Schools with which researchers are associated.
- *Financial Services* is an EPFL department that manages the accounting of the research projects.
- *GrantsDB* is an EPFL internal IT application used to store the general information about a research project, such as the project's status, people involved in it, the planning of finances, etc.

The application process for research funding is separated into three parts:

1) **Building a project proposal**
The process begins with a "call for a project". The Researcher (Tom) first creates a project proposal in GrantsDB with the basic information like scope of the project, initial planning, resources needed etc. He then sends the proposal to the ReO to verify the compliance with the FO requirements. The ReO reviews the proposal, and if needed, they send it to the TTO, for an evaluation and for the patenting of new inventions and technology that will results from the research project. At this point, the researcher (Tom) gets feedback from the ReO and the TTO, and he updates the proposal accordingly. The ReO and TTO check if the proposal is compliant with internal policies.
2) **Validating the project proposal**
The project proposal needs to be approved by the Dean (Jim). Tom is responsible for sending the proposal to Jim. If Jim does not approve the proposal, the process ends.
3) **Finalizing the project** If Tom does obtain the approval, he collects all other needed documents and sends the complete project documentation to the FO. The project documentation varies among different funding organizations. Tom updates the GrantsDB and uploads all project documentation.

After receiving the complete project documentation, the FO informs Tom of their decision (whether the project is accepted or not). Tom then records this decision in GrantsDB. If the project is refused, the process ends. Otherwise, the Financial Services open a fund for the accepted project. With the fund account opened, Tom can start executing the project. The funding application process ends once the Funding Organizaiton informs the researcher of their decision.

III. SEAM Service Model

We begin building a SEAM model with systems identification, based on the services and processes we are modeling. For clarification, we use the term system to refer to any kind of entity: an organization, an employee, an IT system, or an application [7]. In General Systems Thinking (GST), the common definition of a system is "a set of components interacting with each other that creates emergent properties for the observer" [8].

For the systems identification we choose to model systems as

- wholes, denoted as [w], or
- composites, denoted as [c].

By modeling a system as a whole [w] (black box), we do not show the system's components and we focus only on the services offered by the system. When we model a system as a composite [c] (white box), the components and their relationships are shown, so we visualize the implementation of the service, i.e. the process, and understand the responsibility of each system.

A. SEAM As-Is Service Model

In the case-study process description, every actor corresponds to a system in the model, so there is a system for the Researcher (Tom), the ReO, the TTO, etc. Next we look at the relations between these systems and build the systems' hierarchy by using the whole [w] and composite [c] concepts.

There are several possibilities in which the modeler can view these actors and defines the system hierarchy in the model. Each of the following modeling options facilitate differently the design of services:

- Model all actors at the same level of hierarchy. There is no organizational boundary between actors and their systems, and they all work together in the process. This option does not reflect a service offering in any way.
- Model two levels of hierarchy, capturing EPFL's legal boundaries.

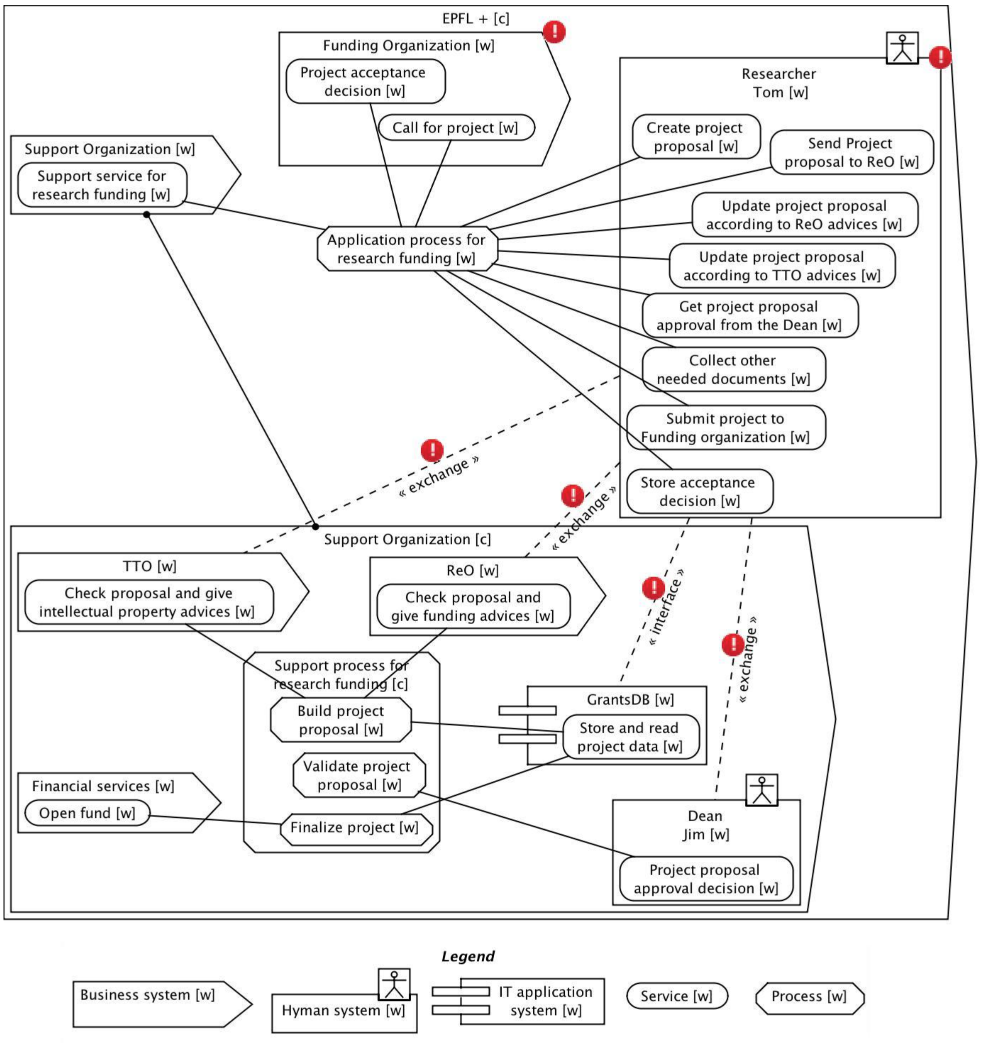
EPFL + [c]
Funding Organization [w]
Project acceptance decision [w]
Call for project [w]
Researcher
Tom [w]
Create project proposal [w]
Send Project proposal to ReO [w]
Update project proposal according to ReO advices [w]
Update project proposal according to TTO advices [w]
Get project proposal approval from the Dean [w]
Collect other needed documents [w]
Submit project to Funding organization [w]
Store acceptance decision [w]
Support Organization [w]
Support service for research funding [w]
Application process for research funding [w]
« exchange »
« exchange »
« interface »
« exchange »
Support Organization [c]
TTO [w]
Check proposal and give intellectual property advices [w]
ReO [w]
Check proposal and give funding advices [w]
Support process for research funding [c]
Build project proposal [w]
Validate project proposal [w]
Finalize project [w]
GrantsDB [w]
Store and read project data [w]
Financial services [w]
Open fund [w]
Dean
Jim [w]
Project proposal approval decision [w]
Legend
Business system [w]
Hyman system [w]
IT application system [w]
Service [w]
Process [w]

Fig. 1. SEAM as-is model of the case study, where exclamation marks (!) signal errors that are discussed in Subsection III-B

  - *Level 1* - World of research funding, containing the FO, as an external partner in the process, and EPFL.
  - *Level 2* - EPFL, containing the Researcher Tom, ReO, TTO, FS, Dean (Jim) and GrantsDB.
- Model two levels of hierarchy that show a support service implementation and offering:
  - *Level 1* - Extended EPFL, denoted as EPFL+, containing a Support Organization, the customer of the process, who is Tom, and the FO, as an external partner.
  - *Level 2* - Support Organization within EPFL, for providing support in the application process for research funding, containing the ReO, TTO, FS, Dean (Jim) and GrantsDB.

We choose the last option that shows clearly the boundaries between the Support Organization for the process, the external partner and the customer (see Fig. 1).

We know the components of the *Support Organization* system, so we model it as composite [c] including the following systems as wholes [w]: the *ReO, TTO, Dean (Jim), GrantsDB* and *Financial Services*. The *Researcher, Tom [w]*, is outside of the *Support Organization [c]*, but inside the EPFL + as a composite. As mentioned, we extend EPFL's boundaries in the model by including the *FO* as part of EPFL; the FO together with the *Support Organization* and the *Researcher*, represent the *EPFL+ [c]* composite system.

In SEAM, the action performed by an actor is modeled as services in the actors' systems as wholes. For example, *create project proposal* is a service of Tom. Another example is the *open fund* service for the *Financial Services* system.

In a SEAM model, we also show processes as actions for the systems as composites. By SEAM, and other definitions [9], a process implements a new service by combining other services. The *application process for research funding* in the *EPFL+ [c]* composite and the *support process for research project funding* in *Support Organization [c]* composite are examples of processes. As for systems, the [w] and [c] letters beside the names of the services and processes denote that the whole and composite concept is used for the actions, thus giving a functional hierarchy. We also choose to see services or processes as black boxes (wholes [w]), or as white boxes (composites [c]). In our example, we only have one process as a white box, the *support process for research funding*, where we model its three sub-processes (the three parts in the description): *build project proposal, validate project proposal* and *finalize project*.

In our model, there are links between services and processes. In Fig. 1 the *application process for research funding* is linked to *project acceptance decision* service, *create project proposal* service, etc.

In Fig. 1 we model both the whole and the composite of the *Support Organization*. The line between these two systems show the refinement from a system as a whole and its service, to a system as a composite and the process, i.e. the implementation of the service. The refinement link is between the *Support Organization [w]* and the *Support Organization [c]*, and it gives the functional refinement from the *support service for research funding* to its *implementation as support process for research funding*.

Dashed lines between systems as a whole show which actors communicate (exchange information, or there is an interface for them) with each other.

In SEAM, there is no explicit order in which these services are executed within the process. SEAM is declarative, and it uses the value of properties or states of systems as a whole to determine which action should be executed at a given time. Here we do not show the properties and their values.

B. SEAM As-Is Model Discussion

SEAM is declarative, so there is no sequence in executing the services and the process flow is not obvious. In Fig. 1 we do not show what defines the order of execution: the preconditions and the postconditions for each action. Adding this would increase the complexity of the model.

An advantage of declarative modeling is that it allows us to concentrate on the purpose of the actions modeled, not the correctness of the execution sequence. The purpose is to have a model with few declarative actions, where the responsibilities of each actor are clearly visible, rather than a model with a detailed sequence of actions.

But people tend to explain the processes and their actions within a sequence. This observation is clearly visible in the model depicted in Fig. 1 from the way Tom's services are formalized and organized. Looking closely at these services we notice problems. In SEAM we make problems clearly visible by adding error signs (note the signs on Tom's system, the dashed lines and the FO).

An example of a problem in the model is that Tom is aware of the support service implementation details (there is no abstraction). This is reflected in the level of details of his services. In the model we see services like "*Send project proposal to ReO*" and "*Get project proposal approval from the Dean*". For the execution of these services, Tom knows that he needs to interact directly with the ReO and the Dean.

C. Perceived Values and Motivations of the Stakeholders

We model the case-study process with the description from Subsection II-A, and we have identified problems with the existing service provider. As we ourselves are researchers, and have experience in the application process for funding research, we have personally encountered the issues discussed in Subsection III-B. Based on what we know of the motivations of researchers and the documentary research evidence we collected from several funding organizations web sites, we drew a mock-up of a SEAM goal-belief model [10], [11]. This model, depicted in Fig. 2, exhibits the actors motivations towards the process for funding research.

Knowing these motivations helps us to understand the value that the actors expect from the service. Through this value, we recognize what a service implementation should do.

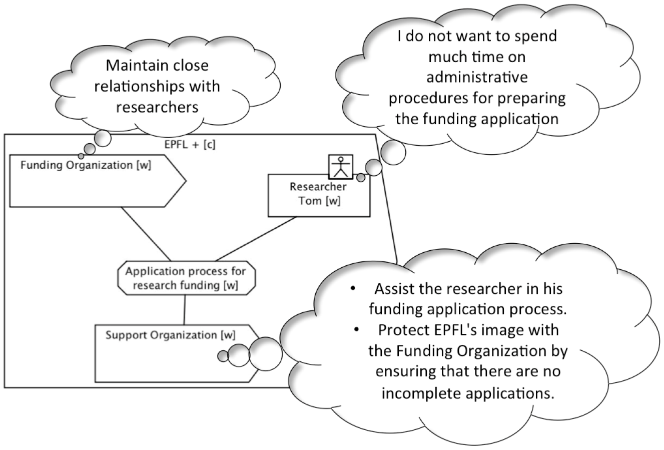
Maintain close relationships with researchers
I do not want to spend much time on administrative procedures for preparing the funding application
EPFL + [c]
Funding Organization [w]
Researcher Tom [w]
Application process for research funding [w]
Support Organization [w]
• Assist the researcher in his funding application process.
• Protect EPFL's image with the Funding Organization by ensuring that there are no incomplete applications.

Fig. 2. Mock up of a SEAM goal belief model for the case study

In the following list, we give a more complete description of actors' motivations, that we could not fit in Fig. 2.

- *Funding Organization* - "We want to have a good image as an organization that supports advancement in knowledge. We want to make an impact on academy and/or industry, so we need to maintain close relationships with researchers."
- *Researcher Tom* - "I care about progress and I have a desire to discover, invent and innovate. I need funding and my research interests match the FO's "call for projects". I enjoy doing research and I do not want to spend much time on administrative procedures for preparing the funding application'.'
- *Support Organization* - "We are here to assist the researcher in his funding application procedure. We need to protect him from accepting prejudicial terms from the FO. Also, we need to protect EPFL's image with the FO by ensuring that Tom does not submit an application with incomplete documentation."

The FO wants to maintain a good relationship with the researcher during the application process for funding. Exposing the researcher to the administrative difficulties of the process, creates a distance between the FO and the researcher. This observation leads to the conclusion that it is in the best interest of the FO to have a proper implementation of support service for research funding at EPFL. In other words, this service brings value to the FO, making the FO one of the service customers.

We derive many details from the service implementation and offering, which lead to a better service model that shows: the organization of the actors of the service implementation, the technical details of the implementation, the identification of new customers, etc.

Our observations from the case study are formalized as proposed value-added service design patterns and applied in the SEAM to-be model.

D. Proposed Patterns for Value-Added Service Design

In following list we give a set of patterns that are used as heuristics for value-added service design. These patterns take into consideration customers' motivations and provide a requisite abstraction of the service implementation to the customers. By applying the patterns, we also *redefine the services* in order to reflect the customers' simplified view of the service implementation.

- Pattern I: **simple interface**
  Provide a simple interface to encapsulate the behavior of the service provider organization.
- Pattern II: **incorporate needed external actors into the service view of the organization**
  Actors involved in a service implementation might be external to the organization modeled. Often their participation in the service implementation might be needed. In such cases, extend the organizational boundary and model a *representative*, or an *interface of the external actor within the organization*.
- Pattern III: **recognize new customers**
  In the as-is, usually one role is given to an actor, so the focus is only on this role. But there can be more than one customer for a service. Understanding the values and motivations of actors helps in identifying multiple roles that an actor has. *Perceiving all the roles* reveals that an existing actor might *become one of the service customers*.
- Pattern IV: **data and process linking technology**
  Actors involved in a service implementation might use different technologies. To resolve this problem in the enterprise's architecture, use data and process linking technology. Data and process linking technologies [12] include software known as *middleware, workflow engines or portals* that provide standardized access to systems and data.

E. From the Proposed Patterns to the SEAM To-Be Model

In Fig. 3 we show the SEAM to-be model of the case study. Here are the changes we introduce that capture service offering and implementation, marked with modified ($\sim$) and added ($+$) sign:

- Application of the **simple interface** pattern:
  In order to facilitate the communication between Tom and the Support Organization to perform *get needed approval*, we choose to use a Web portal. The Web portal is an interface that encapsulates the behavior of the Support Organization. For example, when the regulations change, which demands approval from other actors in addition to the Dean (Jim), the Support Organization manages it, so Tom does not need to know exactly all the places from which he needs approval. The only interaction Tom has is with the Web portal.
- Application of **incorporate needed external actors into the service view of the organization** pattern:
  The Support Organization becomes *Support Organization+* as we add the external partner, FO, within its boundaries. It is to be noted that the *FO* modeled here has a different role than the role it has in the *EPFL+* level. Within Support Organization, FO holds an administrative position that deals only with technical procedures concerning application files, documents, deadlines, acceptance decision propagation, etc.
- Application of **recognize new customers** pattern:
  The *Funding Organization* in *EPFL+* offers a different service, *maintain relationship with researchers*, which reflects the organizations motivations. At the *EPFL+* level, the *FO* is not affected by the technicalities of the funding process. Here, Funding Organization does not play the role of a fund provider, who takes care only of administrative procedures, but becomes a customer because of the close relation with Tom.
- Application of **data and process linking technology** pattern:
  We add a *Web portal* in the *Support Organization+*, with a direct interface to *Tom*. This portal represents the data and process linking technology that we add to the process. Every detail about the communication between different actors, the process workflow and the guidance go through this portal. In this way there is no direct interaction through different interfaces among actors from the *Support Organization+* and the *Researcher Tom*.

After the application of these patterns, we are able to redefine Tom's services. The details that are not relevant to Tom are abstracted. Unlike the as-is situation, this reflects Tom's simplified view of the support process. Here, Tom participates in *create complete project proposal*, where he prepares and modifies the proposal according to the updates he receives. Finally, he is informed about the approval and the acceptance decision.

IV. RELATED WORK

Our related work addresses tools, methods, frameworks, etc., that can be divided based on two main functionalities of service design.

Distinguishing the service offering from its implementation: There are many ongoing works like Archimate (including TOGAF), URBA, SOMA, which look at the service oriented approach. Here, a service is implemented by a process. Mostly, these languages and notations analyze the general service offered before categorizing the specific services. This could become a hurdle to express intermediate services. Thus, there

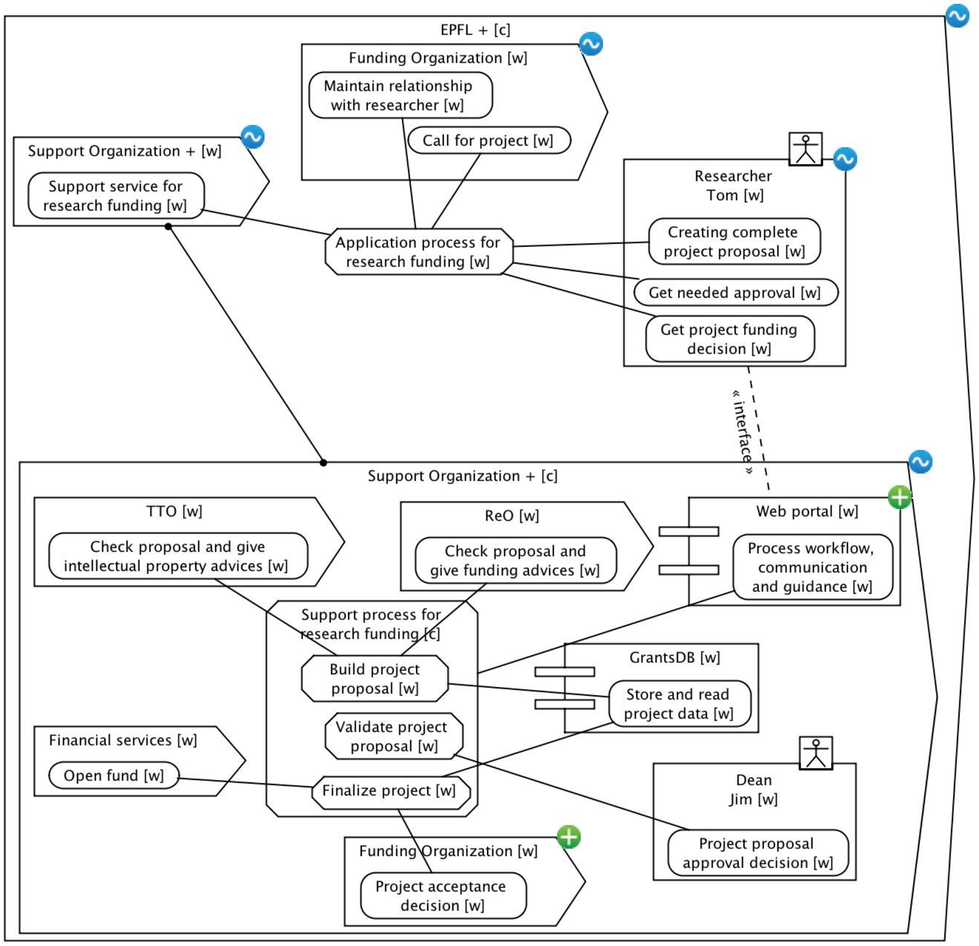
EPFL + [c]
Funding Organization [w]
Maintain relationship with researcher [w]
Call for project [w]
Support Organization + [w]
Support service for research funding [w]
Application process for research funding [w]
Researcher Tom [w]
Creating complete project proposal [w]
Get needed approval [w]
Get project funding decision [w]
« interface »
Support Organization + [c]
TTO [w]
Check proposal and give intellectual property advices [w]
ReO [w]
Check proposal and give funding advices [w]
Web portal [w]
Process workflow, communication and guidance [w]
Support process for research funding [c]
Build project proposal [w]
GrantsDB [w]
Store and read project data [w]
Validate project proposal [w]
Financial services [w]
Open fund [w]
Finalize project [w]
Dean Jim [w]
Project proposal approval decision [w]
Funding Organization [w]
Project acceptance decision [w]

Fig. 3. SEAM to-be model of the case study, where (∼) sign shows what has changed compared to the as-is model, and (+) shows what needs to be added.

is a necessity to express the process that implements the service without losing details or organizational hierarchy.

- Information Technology Infrastructure Library (ITIL) is a set of best practices for IT service management. This includes the entire life cycle of a service from development to release and change management. ITIL also discusses about the best practices to be followed while in a process like incident or problem management which are necessary to provide uninterrupted service.
- In ArchiMate [13], there is service orientation and layering. It has three architecture domains (technology, application, business). Each consists of a process to build a service that is necessary for the higher level.
- In Urbanisme des Systèmes d'Information - EA (URBA-EA) [14], similar to ArchiMate, there are different layers. Each layer implements a set of functions (process) that are necessary to support the higher-level layers.
- Service-oriented modeling architecture (SOMA) [15] defines the elements of the SOA layers and provides modeling, analysis, design techniques and activities for describing services. It uses goal-service modeling to select key services.

Finding patterns for value-added service design: Another challenge of service design is to understand the value to the customer offered by the service. The customer value creation can be understood only if the needs and motivations of the customer are known.

- In i* [16], the models are used to portray actors and their dependencies on other actors. These models are created to understand that an actor needs the help of other actors to achieve his tasks and goal. What may seem as an impossible task for an actor, can become achievable with the tasks of others. This model also helps the stakeholders to understand the use of different IT systems.
- Business Motivation Model (BMM) [17], apart from business plan, also captures and justifies what the business aims at and how to assess the performance and result. Recently, BMM has also been included as a part of OMG specifications.

V. Conclusions and Future Work

In this paper we have shown the challenges faced when defining value-added services. Through modeling a case-study process SEAM, we find ways to implement a service. In a

SEAM service model, there is the systems hierarchy identification and the identification of services and processes within each system. In an as-is model, we show only the systems' behavior and hierarchy, we do not have enough information to contemplate the services provided.

Service customers care more about the value they get from a service, rather than the way the service is implemented by a process. This message is conveyed best by ITIL's service definition given in the introduction. SEAM uses more than one type of model to reason about service modeling. It offers some heuristics to reason about the actors' values and motivations for providing or consuming a service. In a mock-up SEAM goal-belief model, we have shown the resulting motivations and values of the three main actors. In this paper, through a case study modeled in SEAM, we have shown that knowing the motivations and values helps us find and create value-added service design patterns, that are later on applied in the to-be model.

As future work, our objective is to educate people on how to recognize the value to the customer that comes from services. We will apply and validate the patterns listed here on other studies. We will also build a coherent theory behind these patterns and extend the list of patterns.

REFERENCES

[1] P. P. Maglio, S. L. Vargo, N. Caswell, and J. Spohrer, "The service system is the basic abstraction of service science," *Information Systems and e-business Management*, vol. 7, no. 4, pp. 395–406, 2009.

[2] "Information Technology Infrastructure Library," http://www.itil-officialsite.com/.

[3] A. Cartlidge, A. Hanna, C. Rudd, I. Macfarlane, J. Windebank, and S. Rance, "An introductory overview of itil v3," *The UK Chapter of the itSMF*, 2007.

[4] G. Tapandjieva, D. R. Marchetti, I. Rychkova, and A. Wegmann, "Towards the Definition, Implementation and Communication of an IT Strategy: the Case of IT Strategy at EPFL," in *The 8th International Workshop on Business/IT-Alignment and Interoperability*, 2013.

[5] A. Wegmann, "On the Systemic Enterprise Architecture Methodology (SEAM)," in *ICEIS*, 2003.

[6] "QS World University Rankings 2013/14," http://www.topuniversities.com/university-rankings/world-university-rankings/2013.

[7] A. Wegmann, A. Kotsalainen, L. Matthey, G. Regev, and A. Giannattasio, "Augmenting the Zachman Enterprise Architecture Framework With a Systemic Conceptualization," in *Enterprise Distributed Object Computing Conference, 2008. EDOC'08. 12th International IEEE.* IEEE, 2008, pp. 3–13.

[8] G. M. Weinberg, "An introduction to general systems thinking," 1975.

[9] T. Wolff, "Using models to design business processes and services," 2011.

[10] A. Wegmann, B. Bajic-Bizumic, A. Golnam, G. Popescu, G. Tapandjieva, A. B. Saxena, M. Yassaee, and G. Regev, "Requirements modeling in seam: The example of a car crash management system," in *Comparing Requirements Modeling Approaches Workshop (CMA@ RE), 2013 International.* IEEE, 2013, pp. 25–30.

[11] 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.

[12] J. W. Ross, P. Weill, and D. C. Robertson, *Enterprise Architecture as Strategy: Creating a Foundation for Business Execution.* Harvard Business Press, 2006.

[13] M. Lankhorst, *Enterprise Architecture at Work: Modelling, Communication and Analysis.* Springer, 2009.

[14] C. Longépé, *The Enterprise Architecture IT Project: the Urbanisation Paradigm.* Butterworth-Heinemann, 2003.

[15] A. Arsanjani, "Service-oriented modeling and architecture," *IBM developer works*, pp. 1–15, 2004.

[16] J. Gordijn, E. Yu, and B. van der Raadt, "e-service design using i* and e3value modeling," *IEEE software*, vol. 23, no. 3, pp. 26–33, 2006.

[17] Object Management Group, "Business Motivation Model, Version 1.1," http://www.omg.org/spec/BMM/1.1/, 2010.