Journal of Service Science Research (2015) 7:55-96
DOI 10.1007/s12927-015-0003-2

A Service-Oriented Framework to the Design of Information System Service

Valter Castelhano de Oliveira, José Reinaldo Silva

Received: 30 January 2015 / Accepted: 25 August 2015 / Published: 31 December 2015

ABSTRACT

The beginning of this century is marked by a paradigm shift due to the move of the production focus from goods-dominant to a service-dominant. At the same time, manufacturing automation and integration are undergoing changes, which open the possibility for classical model oriented products to be replaced by service models, supported by cognitive information systems. This paper analyzes a proposal to achieve a sound design process for service systems, which follows the model driven tendency. In fact, the aim is to bring together practical and formal approach, and therefore, to propose a good design discipline based on SOMF (Service Oriented Model Framework). Based on this model driven approach a new environment were developed which supports elicitation, modeling and requirements analysis supported by semi-formal methods (SOMF and UML) and formal methods (by using SysML and Petri Nets). The proposed method is applied to a case study based in an urban Smart Grid.

KEYWORDS

Service Science, Information Systems, UML, SysMl, Petri nets, Smart Grid

Valter Castelhano de Oliveira
Mechatronic Department, University of Sao Paulo
e-mail: vcoliv@usp.br
José Reinaldo Silva
Mechatronic Department, University of Sao Paulo
e-mail: reinaldo@usp.br

SoSS THE SOCIETY OF SERVICE SCIENCE

Springer

1. INTRODUCTION

The beginning of this century was marked by a potential paradigm shift in the modeling and design of production processes, migrating from a goods-dominant to a service-dominant approach (Lusch et al. 2008). In fact, this reflects a change in the economy, business processes (BP) and pervasive automation systems, since this change is happening in parallel with the diffusion of service orientation as a business reference (Kim & Nam 2009).

The delivery of products is being replaced by the search for goal-oriented solutions, planned and assisted by information systems, and based on resources, some of which depend on "old" products. Such a tendency characterizes a movement from goods dominant logic (G-D logic) to service dominant logic (SD-logic) (Spohrer et al. 2008; Lusch et al. 2008; Vargo & Akaka 2009).

Service systems are the support for S-D logic, characterized by a dynamic configuration of people, technologies, organizations, and information sharing, which allow new and former suppliers to create and deliver value to its customers. Still, the key concept associated with industrial service systems is that they interact to co-create value involving suppliers and customers (Spohrer et al. 2007; Maglio et al. 2009). Therefore, service systems are becoming the focus for integrative and hybrid deliverable artifact composed of hardware, software and people, generically called "service."

However, the automation and abstraction provided by these systems depend on technological innovation as well as on the solutions adopted for the management of resources, information, and restrictions inherent in the application. Thus, classical techniques used to develop information systems (IS) should be revised to adapt to new challenges posed to the development of these new systems.

The convergence between service systems and IS occurs precisely due to IS characteristics of flexibility and integration-a key issue to services associated to innovation and automation. Such features suit service systems on their need to identify its functionality and goals in a heterogeneous environment, serving to multiple purposes and to a wide range of users and stakeholders (Stair & Reynolds 2010).

We are facing a paradigm shift from a focus on goods to a focus on service, and as it happens several times in history, economy and engineering are the basis for this shift. Together, these forces leads to a demand for new system design techniques entirely focused on the functionality (the goals) of both the artifact and the service. Thus, mechatronic design-a clearly interdisciplinary approach-is a reference model for the new world of service engineering.

The importance of information technology services is growing very fast in the global economy, which favors to focus more attention to IS, especially on the theoretical basis to integrate managerial and technical knowledge (Bardhan et al. 2010). Concerning the early phases of the development processes, IS development requires the participation of people with different profiles and expectations: the end user expects an outcome that meets their daily needs and facilitates their work, funders expect an adequate return on their investment, and the development team hopes to develop an artifact of quality in the stipulated time and cost and meets the real needs of users and stakeholders (Leffingwell & Widrig 2000). Even restricted to these three classes of agents-which would be a good approximation-the requirement engineering process would still be a key issue to get good service systems.

This paper analyzes some proposals to compose a design process for service systems following some of the requirements depicted above and aiming to produce automated integrating systems. A framework for developing information systems for automated service is proposed, called SoftDiss (Service-Oriented Framework to the Design of Information System Service). SoftDiss is an environment that supports elicitation, modeling and requirements analysis, supported by semi-formal methods (SOMF and UML), and some formal modeling representations such as SysML and Petri Nets. The proposed method is applied to case study based on an urban Smart Grid system.

2. DEVELOPING SERVICE SYSTEMS

Basic concepts of Service Systems are well defined in (Spohrer et al. 2008; Lusch et al. 2008; Vargo & Akaka 2009). Service Systems were already considered in its recursive composition to aggregate value and achieve an economic impact (Maglio et al. 2009; Spohrer et al. 2007; Cambridge 2007), which is also a key concept to architecture of systems, based on information systems proposed in this paper.

Service science or SSME (Service Science, Management and Engineering) is associated with the study of service systems design (Lusch et al. 2008; Spohrer el al. 2008; Cambridge

2007). As a discipline in training (Ng & Maull 2009; Zhao & Perros 2009; Li et al. 2007), the service science requires a thorough and systematic study that makes it possible the creation of a sound framework, appropriate to be applied in several areas of knowledge.

However, to produce methods for Service Engineering a formalization of Science Service is demanding (Moussa et al. 2010). A basis for that is the use of S-D Logic (Lusch et al. 2008), even if incipient, provides a basis for a formal approach to service science (Vargo & Akaka 2009; Maglio et al. 2009).

In the work of Barile & Polese (2010) the search for a Science Service formal approach goes through the field of general systems theory (Bertalanffy 1950; Boulding 1956), exploring the relationship between the proposed approach based on VSA-Viable System Approach-proposed by Beer (1984)-and recent advances in the area of Services Science, called Smart Service Systems. VSA is an interdisciplinary theory applied to observe complex phenomena, based on systems theory that focuses on the relationships between socio-economic entities to search viable conditions for interaction between them. The main result is that the VSA provides valuable information to the design and management of systems for intelligent services, especially regarding harmonization, governance, and co-creation.

Kim & Nam (2009) proposed a theory of service systems based on a systematic procedure to understand the nature of services and built an integrated theory of service systems that leads to innovation and stimulate productivity, by providing foundations to design, production, delivery, operation, maintenance, monitoring and maintenance of service systems. They identified influential agents in a service system as: customer, product, BP, participants, information and technology.

The interaction between these agents is complex and derives from the knowledge of stakeholders, from the intangible exchange of knowledge that arises in the integrated network of participants and resources to create customer value. Goals of service systems should be analyzed into levels of quality (perceived by the customer), productivity (attributed to providers of service) and innovation. The contribution of this work comes from studies in the areas of quality, productivity and service innovation, looking for a design process that leads to good service systems.

Bardhan et al. (2010) propose a framework to evaluate the main research lines in SSME, with a multidisciplinary analysis, including IS, computer science, economics, finance, marketing, operations management and supply chain management. As a result of the analysis the authors obtain a comprehensive coverage, with interpretation of the main issues raised; initial theoretical perspectives for further research; applications to better understand the innovation-oriented services, and to position service science as a key research area in IS.

Ostrom et al. (2010) compiled the opinion of academics working in related disciplines, and executives connected to at least a thousand of small, medium and large companies. This task force helped to set a list of the ten research priorities for the development of Science Service, grouped in three areas of business:

- Strategic Priorities: foster the dissemination and growth of services; improving the quality of life through services; and is able to create and maintain a culture of service.
- Development Priorities: stimulate innovation in services; improving the design of services; improve the service networks and value chains.
- Execution Priorities: to make effective the brands associated with the sale of services; improving the service experience through joint creation with the customer; and measure and improve the service value.
- Permeate of all priorities, using technology to leverage the service.

Stanicek & Winkler (2010) presented a framework for modeling information associated with service systems based on semantic conceptual modeling and goal-driven approaches. Normally, the modeling of service systems should be focused on relations rather than on objects that compose the system. The modeling registers the needs to be attended-or problems to be solved by the system-and also requirements determining attributes, capabilities, features or qualities of the system. Refinements made on the objectives or goals of a service system can be documented graphically in a service breakdown structure (SBS), which introduces a sub-service as part of the service delivering. This modeling approach provides a conceptual semantic communication language common to both the provider and to the service consumer.

However, the most complete proposal (even if semi-formal) is the SOMF (Service Oriented Modeling Framework) proposed by Michael Bell (2008). SOMF is a software development method based on a modeling language and an anthropomorphic holistic, applicable to various levels of abstraction required by a system service life cycle, from the business process view to the realization of the system design. It offers (to analysts, architects, developers, modelers and managers) two lines of action during the process of software development: one based on what should be done (what to do) and on how it should be done (how to do). Both ways are directed by a modeling language that support a service-oriented software apprentice based on different modules: i) a general holistic view of all the software entities of the organization, such as legacy systems, services, infrastructure or BP; ii) a module of analysis, design and architecture driven models with reusability capacities; iii) a module with portfolio management practices to service life cycle; iv) a module for notation easy to learn and use; v) a defined set of modeling viewpoints: conceptual analysis, design, business integration and architecture.

Therefore, all elements to a complete life cycle are present up to the extermination of the service. That is, services (similar to products) are also subject to changes and upgrades, or eventually extinction, which means a need to review its requirements and return to the design process. This cyclical and iterative process is a key issue in the lifecycle of the service.

Also, SOMF offers a wide range of disciplines and good practices toward a successful model-driven development of services (especially those based on information systems). In fact, there are six service-oriented modeling disciplines, grouped into three desktop environments: a) Conceptual Environment, containing the disciplines of Service Conceptualization and Conceptual Architecture; b) Environmental Analysis, containing the disciplines Service Discovery and Analysis and Business Integration; c) Logical Environment, containing the disciplines Service Design and Logical Architecture. Each environment offers specific and interactive representation that contributes to the general model.

For all these reasons SOMF currently a good approach to systems service based on information systems and therefore was taken as a basis for our proposal. On the other hand, the development of Service Engineering would demand more than best practices, even if they work nicely to a great variety of practical cases.

2.1 Information System Service

Service systems became the focus in business integration and are spreading its influence to other areas bringing technological innovation, automation and efficiency. However, the automation and abstraction that these systems provide depend on planned solutions to manage resources, information and restrictions inherent to the application. Therefore, classical techniques used to develop information systems must be revised to adapt to new challenges posed by new science services (Spohrer et al. 2008; Chesbrough & Spohrer 2006).

New information systems must meet more sophisticated requirements allowing the integration of several organization components, with more elaborate functions, collecting information from different viewpoints and from the BP (Davenport 1993; Marshall 2000; Aalst 1998; Aalst & Hee 2002). On the other hand, it must keep always the focus on the customer, delivering the information in the right places and to the proper people.

Critical objective of service systems is to develop sound design which allows the implementation of more complex systems with a good level of flexibility and integration, serving multiple functions and a wide range of users and stakeholders (Stair & Reynolds 2010).

Generally, early phases are the source of great part of problems in the development of information systems although the consequences are frequently detected in other stages of development. Paradoxically, the initial stages are less costly because they do not require large investment in programming or resource components. These early steps include the phase of requirements engineering, the most important in the information systems development process (Kotonya & Sommerville 1998). Notice that the cost to repair a requirement problem, discovered when the system is already in production can be up to 500 times greater than if this same problem was detected and treated during the requirements phase (Carr 2000).

According to Kotonya & Sommerville (1998), the goal of requirements engineering is the validation of requirements by stakeholders (individuals or organizations that are benefited by the system). However, discovering mistakes and testing requirements is a long activity, involving a heterogeneous group of people to seek problems, omissions and ambiguities in the requirements document, thereby generating a final version, which is called a specification. Despite all this, the main problem is precisely the absence of a reference model to be used as a basis for validating requirements-which would give a certain degree of quality to these specifications (Oliveira & Silva 2011).

Since there is not an ideal method for requirements engineering, in this work we choose to use Volere, a practical requirements engineering method proposed by Robertson & Robertson (1999) and the modeling discipline Naked Objects (Pawson 2002). Naked Object encourages the specification of systems from behaviorally complete objects, keeping the data and behavior of grouped objects integrated, and seeking to identify the key business objects.

Information Systems Service (ISS) should match the required features and attributes of services, and the development and maintenance process is directly linked to this requirements. Maintenance and traceability of system requirements must be performed in a heterogeneous environment, serving multiple functions and a wide range of users and stakeholders-represented by people, institutions and other systems-all interacting with the system, where each agent has a different viewpoint of the system. Thus the development of an ISS must consider the diversity of its viewpoints (Leite 1996; Kotonya & Sommerville 1996; Sommerville et al. 1997) and it has to ensure traceability (Letelier 2002; Gotel & Finkelstein 1994; Tang et al. 2007) to associate interaction with requirements throughout the development (and maintenance) process.

According Kotonya & Sommerville (1998), viewpoint is a collection of information presented through a particular perspective on a particular issue, environment, domain or system. Each of these perspectives can be associated with system users, other systems, stakeholders, and also with the system development team. Generally, the information associated with each viewpoint are incomplete, as well as the system requirements obtained from the integration and combination of different viewpoints, leading to the need of a more sophisticated requirement engineering work to detect and solve conflicts and inconsistencies.

Leite (1996) identified three main contributions to an effective improvement of the production of information systems: recognizing that multiple views must be considered throughout the development process of the system; recognizing that concept is important to representation specifications (which supports consistency analysis and model verification); and the use of viewpoints as services to be provided by the system.

Requirements traceability is the ability to describe and follow the existence of a system requirement, since its inception to retirement system, through the various stages of development, specification, implementation and maintenance (Gotel & Filkelstein 1994). Requirements traceability can also help to ensure a matching between stakeholder requirements (specially those connected to BP) and the artifact produced during the software development process (Letelier 2002).

However, the effectiveness of traceability in practice can be questioned because there is no detailed guidance on the types of information that must be met, or the context in which such information should be used. Also, there is not a consensus about the semantics for connections between different abstraction levels such as specification and implementation (Letelier 2002).

Advances in computing lead to a change in the design of systems approach, from document-driven to a model-driven process. The introduction of models improves the quality of system specifications where target information is captured by relations connecting agents and can be reused in multiple diagrams. Moreover, traceability between different levels of abstracttion can be defined explicitly in the model as the relationships between the elements (Pearce & Hause 2012).

OMG Model Driven Architecture-MDA (http://www.omg.org/mda/) is based on MOF (Meta-objet facility, http://www.omg.org/spec/MOF/2.0/) and incorporates the importance of using models in information systems design. The model-driven process should start with a very abstract and conceptual model of the target system and generate more detailed models by successive transformations, to the implementation level. Each new model possesses a higher degree of formalization, reducing ambiguity.

The models generated in the process are: Computation Independent Model (CIM), with a higher level of abstraction and describes the area and the requirements on the system; Platform Independent Model (PIM), defined from the CIM and independent of technology and platform, but representing business where the system falls, Platform specific model (PSM) are specific models that take into account the technology used to implement and could lead to a different PIM PSM (Kent 2002).

OMG MetaObject Facility-MOF (http://www.omg.org/mof/) is a foundation environment proposed by OMG (Object Management Group) where models can be exported and imported from different applications, transported across a network, stored and retrieved in a repository and rendered into different formats, including OMG XML Metadata Interchange-XMI (http://www.omg.org/spec/XMI/).

According Estefan (2007), a methodology Model Based Systems Engineering (MBSE) is a set of processes, methods and tools used to support systems engineering in a context-based models. OOSEM (Object-Oriented Systems Engineering Method) (Friedenthal 2009) is an object-oriented approach (Raumbaugh et al. 1991) for systems engineering, developed by INCOSE, which together with SysML (OMGSysML 2012) composes the MBSE (Model-based systems engineering) (Pearce & Hause 2012).

The methodology OOSEM integrates a model-based approach and SysML to support the specification, analysis, design and verification of systems (Estefan 2007). OOSEM uses concepts of object orientation in conjunction with the methods of traditional systems engineering (top-down), supporting the specification of flexible and extensible, enabling accommodate evolving technology and changing requirements. OOSEM also intended to facilitate integration with object-oriented development of software, hardware development and testing. The activities and tools OOSEM modeling process are described below:

- Analyze stakeholder needs: capture the current state (as-is) business systems, its limitations and possibilities for improvements. The result of this analysis is used to derive the business requirements of the service system (to-be).
- Define system requirements: specify the requirements of the systems that support the needs of the mission. The system is modeled as a black box, which interacts with external systems and users represented in the enterprise model.
- Define logical architecture: decomposition and partitioning the system into logical components that interact to satisfy the system requirements.
- Allocate the components in the architecture: describes the relationship between the physical components of the system including hardware, software, data and procedures.
- Optimize and evaluate alternatives: in other activities OOSEM invoked to optimize and refine the candidate architectures.
- Validate and verify the system: to check whether the system meets the specification requirements and validate the requirements that meet the needs of viewpoints.

The ISO/IEC 15288 is a worldwide standard for the lifecycle of engineering processes and software systems and defines an environment (framework) process to be applied to a system throughout its life cycle, including definition and requirements analysis, architecture design, implementation and verification (Haskins 2006).

The ISO/IEC 15288 is organized into 4 groups of processes: enterprise processes are focused on the organization's ability to achieve and sustain the system; agreement processes are used to establish agreements between companies in the acquisition of goods and services; technical processes are used to establish the requirements for the creation of the system and to sustain its operation during the entire life cycle; and the project processes, including planning, resource allocation, control, risk treatment management and communication of project creation and evolution of the system.

It should be normal to wonder if the better approach to Service Engineering would not be based in a well-connected set of diagram that could be transformed in a standard, but maybe based on the best experience of those who practice the development of service systems. In that case, SOMF would be a good candidate, at least to services that are provided for information systems.

A different approach is looking for a formal approach to Service Engineering, even to service information systems.

2.2 Formal Specification of Systems

Concerning (Information) Service Systems, it is important to treat the early phase of the process and to keep in mind the such systems should attend to several and different viewpoints, associated to distinct classes of clients and stakeholders. Besides being a characteristic of any software this is a key issue in service systems.

However it is also important to envisage the anticipation of the formalization, which can be achieved by introducing a formal representation in requirements specification, by introducing new methods of requirements analysis or even methods to balance BP and functional requirements (Oliveira & Silva 2011; Khadraoui & Feltus 2012). In the current development approach that anticipation could be achieved by a process where specifications are represented in a semi-formal language such as UML, and transferred later on to SysML (a formal representation but with limitations in simulation and validation) and finally to Petri Nets or another state-transition formalism.

SysML (System Modeling Language) is the result of a join effort to improve systems design, involving the development of UML version 2 (Kobryn 2004; OMGSysML 2012), an initiative shared by OMG (Object Management Group) and INCOSE (The International Council on Systems Engineering) to create a system development process based on successive change of representation. The emerging SysML representation is based on a set of diagrams easily interpreted but with strict syntax and semantics, suitable to a formal specification of models. However, SysML has limitations regarding the implementation and mathematical analysis of their models, limiting the ability to analyze and verify requirement specifications.

Through a graphical representation, Petri nets offers greater potential as a language of communication among professionals from various areas, and provides the mathematical formalism suitable for the methods of analysis (Aalst 1998).

In 1962, Carl Adam Petri presented the theory of Petri nets in his doctoral thesis. Nowadays, Petri nets are used as a general structural modeling representation for a great diversity of systems, which can also do the modeling and analysis of information flow. Its basis evolved since 1962 and several extensions appear up to 2004 where a standard were proposed in the ISO/IEC 15909 project (Hillah et al. 2004).

A Petri net is a bipartite graph, non-null, targeted, formed by two types of components, called transitions and places (Murata 1989). The performance of an action is associated with pre-enabled conditions, i.e. conditions of system state variables-stored in locations-which allow to "*firing*" of the given transition. After the completion of the transition occurrence (firing), post-conditions are changed by marking output places. Places are graphically represented by circles while transitions are graphically represented by rectangles or bars. Actually, places and transitions are the vertices of the Petri net supporting graph, and are connected by directed arcs.

Petri net is formally represented by quintuple PN = (P, T, F, W, M_0), were (Murata 1989):

- P = {p1, p2, ⋯, pm} finite set of places;
- T = {t1, t2, ⋯, tm} finite set of transitions;
- $F \subseteq (P \times T) \cup (T \times P)$ set of arcs;

- W: F $\rightarrow$ {1, 2, 3, ⋯} is a weighting function;
- $M_0 \in$ M: P $\rightarrow$ {0, 1, 2, 3, ⋯} is the initial marking;
- $P \cap T = \varnothing$ and $P \cup T \neq \varnothing$.

Petri nets have been widely studied (Murata 1989; Desel & Reisig 2004) and applied to the modeling and analysis of discrete event systems, where sequences of activities, parallel and competing, are combined to represent processes. Classical Petri nets allow the modeling of states, events, conditions, synchronization, parallelism, selection and iteration, but do not provide the necessary resources such as data modeling and processing times, to describe real problems, often complex and extensive. To solve these problems several extensions were proposed, introducing color, time and hierarchy which characterizes a High-Level Petri Net (Aalst 1998; Smith 1998):

- Color: characterizes several types of markings representing objects such as resources, goods and people (Jensen 1998);
- Time: used to model real-time systems, describing its temporal behavior;
- Hierarchy: precise specification of BP tends to be large and complex, involving the use of hierarchical structures in the form of subnets.

An open source development environment GHENeSys (Silva et al 2008), for analysis and design requirements, workflow, information flow and production systems, was designed and developed in DesignLab (http://dlab.poli.usp.br) of the Polytechnic University of São Paulo (POLI-USP), from the ideas presented in Miyagi (1988), Silva and Miyagi (1995, 1996). Designed as an extended object-oriented net, it introduces mechanisms based on hierarchies combined with a structured approach and fits the Petri Net Standard IEC/ISO 15909 (Hillah, Kordon & Petrucci 2004). GHENeSys environment has the potential to become a unified tool for the representation of classic and extended Petri nets as well as high-level Petri Nets. A GHENeSys net is formally defined by a quintuple G = (P, T, F, W, M_0):

- P = {p1, p2, ⋯, pm} finite set of places that is divided in two disjoint groups B and B’, P = B ∪ B’, and B ∩ B’ = ∅, called boxes and pseudo-boxes;

- $T = \{t1, t2, \cdots, tm\}$ finite set of transitions;
- $F \subseteq (B \times T) \cup (T \times B)$ are conventional arcs;
- $L \subseteq (B \times T)$ are enabling or inhibitor gates;
- $K : P \rightarrow N+$, is the weighting function;
- $M_0 \in M: P \rightarrow N$, initial marking.

The introduction of pseudo-boxes is a key issue to allow the representation of observable but incontrollable events. Gates are also important to represent the flow of information but are restrict to have only pseudo-boxes as an origin. That is important to preserve the linear algebraic approach to these nets.

Invariants are fundamental algebraic characteristics of Petri nets and are used in various situations, such as checking liveliness, limitation, proof of ties, and others (Murata 1989). They can now be associated with requirements analysis in the proposed Service Engineering. Invariants represent conservative and repetitive components of a Petri net, allowing the identification of sets of places and transitions that cannot be achieved due to the structure of the network, finding important to analyze the behavior of the system represented by the network.

3. SOFTDISS-SERVICE-ORIENTED FRAMEWORK TO THE DESIGN OF INFORMATION SYSTEM SERVICE

This section presents the practical contribution of this work: a Service-Oriented Framework to the Design of Information System Service, or SoftDiss. SoftDiss is an extended software environment that could be added to existing software environments (the current version is being implemented in the Enterprise Architecture). The main idea is to offer support to the development of (information) service systems (ISS) in the early phases, including elicitation, modeling and requirement analysis by using a dynamic and cognitive configuration of technologies, people, external systems, and also information shared between these elements. The main goal is the realization of a chain of services that meets distinct viewpoints and leads to a formal specification of the ISS.

3.1 SoftDiss-Systemic Structure

SoftDiss is a service system itself that assists service systems developers in the modeling and design initial process to reach a formal specification of the target system.

Among the basic assumptions, SoftDiss assumes that requirements are divided into BP and basic requirements (Oliveira & Silva 2011; Khadraoui & Feltus 2012), covering all distinct viewpoints. The requirements are arranged to compose models of the service system in different levels of abstraction and in different levels of formalization ranging from informal specifications in a semi-formal stage, until a formal requirement specification in a more formal language, such as SysML. BP are strictly linked to stakeholders, while others may meet the requirements viewpoints of various classes of agents, including users, managers, etc.

Figure 1. presents the systemic structure of SoftDiss, where the input is a request to solve a given design problem and the output is the formal specification of the ISS. SoftDiss environment has four environments: Requirement Environment, Design Environment, Formal Environment and Managerial Environment. All environments are service-oriented.

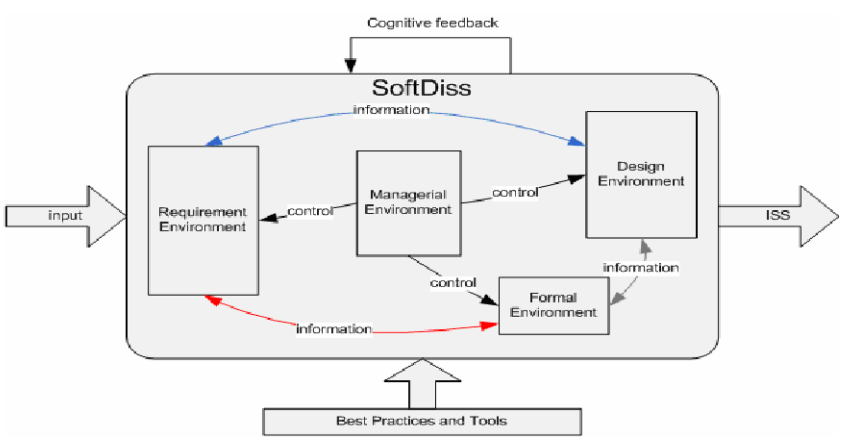

Figure 1. SoftDiss Systemic Structure

The Requirement Environment (ReqEnv) offers services to store, organize and enlighten the viewpoints considered in the requirements specification of the ISS. Therefore, ReqEnv performs activities that support different sets of viewpoints, as BP, user requirements and reference models, across disciplines and methods to maintain traceability and standardization of various sources of requirements. Basic analysis methods are applied in this environment such as the matching between BP and basic requirements (Oliveira & Silva 2011; Khadraoui & Feltus 2012).

The Design Environment (DesEnv) identify and analyze the services necessary to solve the proposed problem and synthesize a formal specification of the ISS, with various levels of abstraction-oriented representation of models, matching the informal and semi-formal representation of the early stages, and favoring communication with those agents involved to validate the process.

The Formal Environment (ForEnv) provides services associated with the consistency of validated requirements, analyzing its dynamics and looking for conflicts. The unified specification then goes through a verification process (not in the scope of this work) or through simulation in a running representation (Object Petri Nets, or any other).

The Managerial Environment (ManEnv) provides management services activities is a kind of supervisory system that controls the information exchange in SoftDiss, supporting planning and control of activities, allocating resources and enabling the flow of information between different environments. ManEnv focuses on management activities of the specification process and system modeling, operating synchronously with activities of other environments.

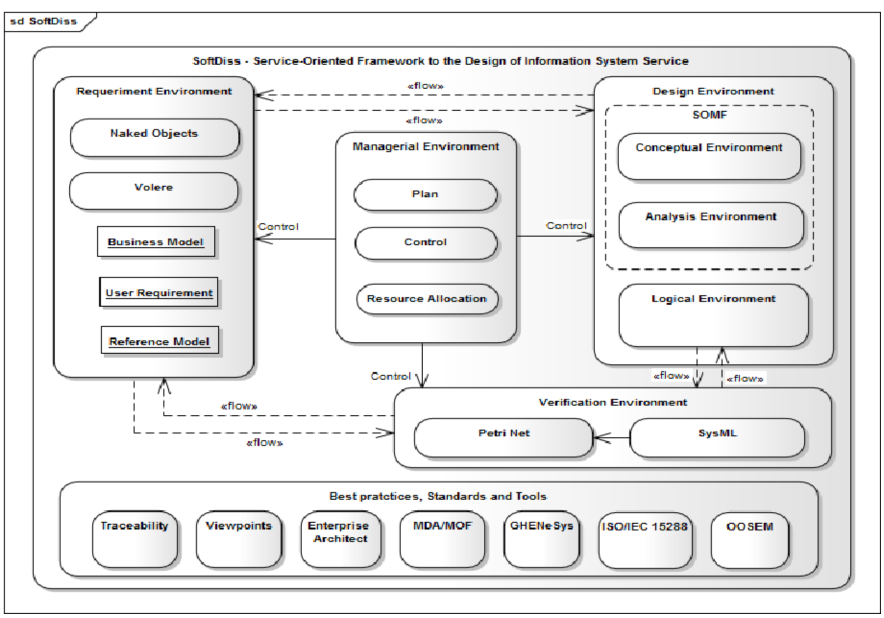

Figure 2. SoftDiss

Figure 2 shows a detailed view of SoftDiss with its internal environments, information flows and control, and identification techniques, external systems, tools and best practices that support the service-oriented environment for specification ISS.

The result obtained by SoftDiss is an intricate set of models representing the various levels of abstraction resulting from both the process of requirements formalization and harmonization among viewpoints target service system. Model Driven Architecture (MDA) and Meta-Object Facility (MOF), both proposed and maintained by the OMG, support the maintenance and processing of various models in SoftDiss. The software tool Enterprise Architect (EA-http://www.sparxsystems.com/) provides resources to support the development, registration and documentation of models.

Next sections will discuss in more details each of SoftDiss environments and the relationship among them, as well as some examples of representations and models used.

3.2 Requirement Environment-ReqEnv

Requirements viewpoints are the input to ReqEnv and can be presented in various types of representation and modeling languages. In the ReqEnv each viewpoint is transformed into a UML object-oriented representation, following the Naked Objects (Pawson 2002) modeling discipline, and Volere (Robertson & Robertson 2006) requirement representation. Both processes were introduced in the general environment Enterprise Architect (EA).

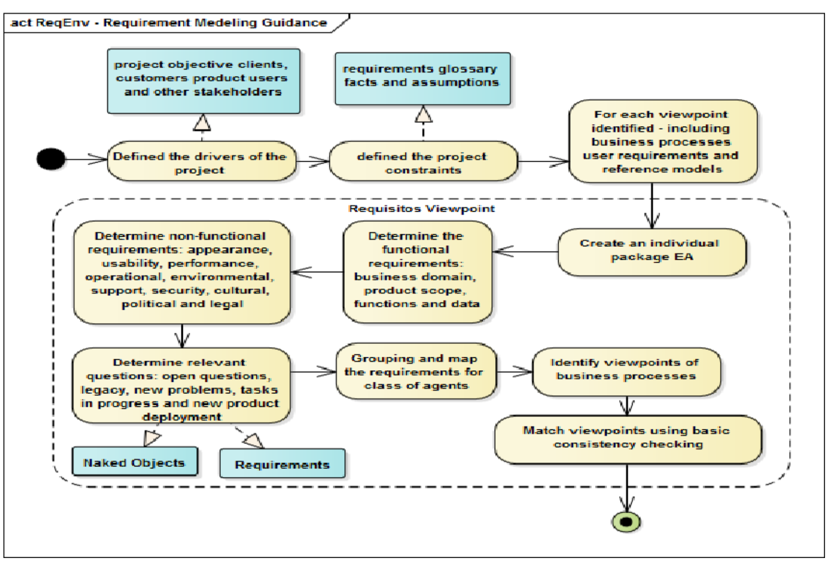

Figure 3. Requirement Modeling Guidance

Figure 3 describes the modeling process in the ReqEnv where the various requirement viewpoints are identified and stored as UML diagrams to be analyzed preliminarily making a distinction between basic requirements and BP.

In the modeling process (Figure 3), first defined the drivers of the project (project objective clients, customers product users and other stakeholders), and project constraints (requirements glossary facts and assumptions). For each viewpoint identified-including BP user requirements and reference models-a package is created in the EA. Following, functional and non-functional requirements are determined and relevant issues are recorded in the EA requirements diagram (Figure 4).

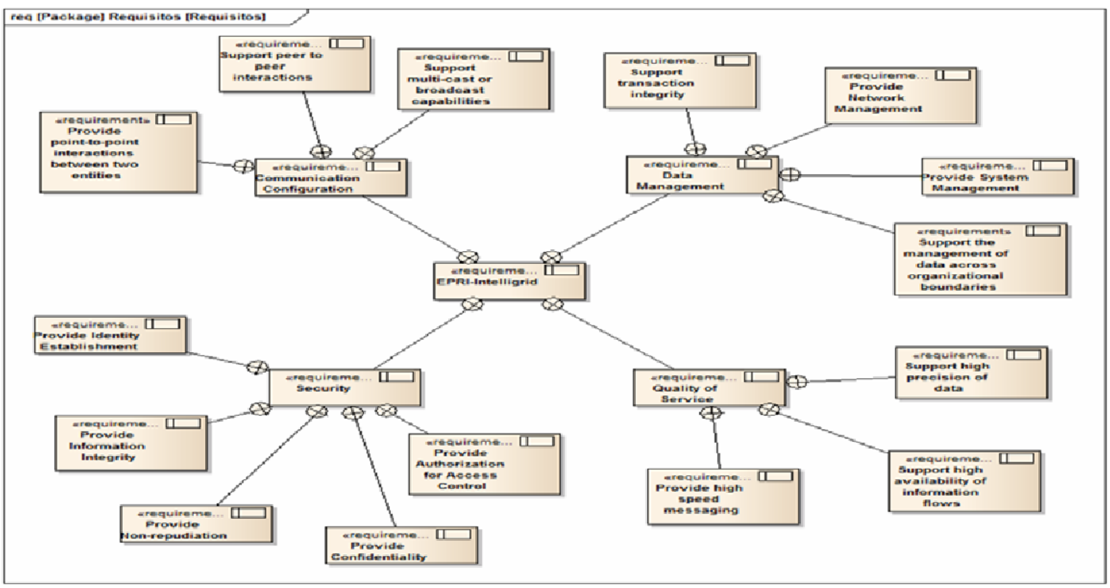

Figure 4. Viewpoints requirements

All requirements are analyzed and identified the following the Naked Objects (Pawson 2002) discipline to be represented in an EA object diagram (Figure 5).

ReqEnv provides service support to treat requirements, representing all viewpoints in a very suitable object-oriented UML, based on the Naked Objects modeling discipline. Being a service-oriented environment, ReqEnv participates in a service chain, with customer-focused operant resources allocated for value co-creation. The operant resources are resulting from the application of the methods processes and techniques of Naked Objects and Volere. The customers of ReqEnv are the other environments working within the co-creation requirements for providing feedback and direction to trading and convergence of requirements along the various viewpoints.

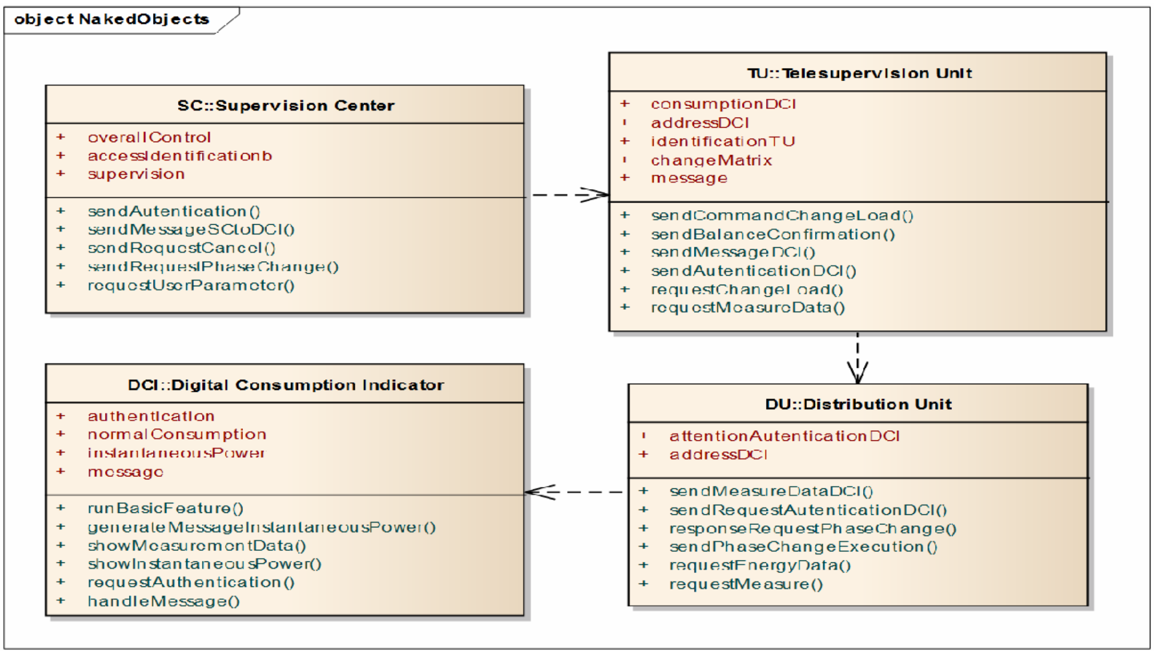

Figure 5. Naked Objects

Requirements organization are performed in the following steps:

i) First of all, requirements are grouped and mapped to a class of agents. It is not possible to have a group of requirements which is not mapped to its respective viewpoint agent;

ii) Second, a special set of viewpoints are identified as the BP (generally mapped to stakeholders);

iii) Compatibility checking is performed using basic consistency checking, that is, the same requirements would be found in different viewpoints even if a metaphoric mapping of terms were necessary (Oliveira & Silva 2011).

The result of this phase should be a direct approval or a report of some difficulties in the consistency checking, which will be stored and made available to the management environment (ManEnv).

3.3 Design Environment (DesEnv)

The DesEnv is basically inspired in SOMF and is composed of a set of internal environments that together allow us to identify, analyze, specify, record and document services associated with the solution of the problem provided at the input of SoftDiss meet the requirements of viewpoints and result in formal specification of ISS. The DesEnv has several levels of abstraction-oriented representation models being informal and semi-formal in the early stages favoring communication with those agents involved, the requirement process and that are supposed to validate the service solution.

The DesEnv is comprised of three internal environments (Figure 2), ConEnv (Conceptual Environment), AnaEnv (Analysis Environment) and LogEnv (Logical Environment).

In ConEnv all requirements elicited from different viewpoints provided by ReqEnv are used as input for defining the conceptual architecture of the identified services. A proper combination of these basic services would be result in the service system that is the solution of the design problem.

The AnaEnv transforms the conceptual services in analytical services, using the problem domain knowledge to validate architecture of the basic services that can provide conceptual solutions for the design. Case a plain compositional service is not found the AnaEnv should return the best approximated solution and report the solution and the gap to the ManEnv. An intelligent solution to identify the nature of the gap is not developed yet. Such tool would be integrated in the ManEnv to make the tracing to the requirements (ReqEnv) to identify which viewpoint is not complete-or at least to get a good guess on that.

The LogEnv transforms analytical services-that is, compatible and harmonic basic service solutions-in more tangible elements allowing to depict relationships in the basic service structure to achieve an optimized service solution, that is, one without duplications or unnecessary expenditure of resources. Also in this tool the general solution should be transferred to a formal representation where the candidate solution could be formally verified or simulated.

The Service Oriented Modeling Framework (SOMF) (Bell 2008) provides practical processes and disciplines include in the initial stages of DesEnv and represented by two of the mentioned internal environments: ConEnv and AnaEnv. In this step the services are identified, conceptualized, analyzed and integrated with business needs-even if a matching procedure, like the one proposed in the ReqEnv is not provided.

SoftDiss aggregate contribution to the environment ConEnv and AnaEnv, originally proposed by SOMF, through the organization mapping and traceability of viewpoints offered by ReqEnv, version control, and interactive documentation view of services offered by the ManEnv.

In the LogEnv, the final stage of the Design Environment as proposed in this work, the semi-formal representation (quite the same one provided by SOMF) is transformed into a formal representation in SysML with accurate representation and free of ambiguities. This representation is runnable, that is, it could be simulated in this representation. In the future we propose an extension of SoftDiss to include a translation of the design solution to Petri Nets in order to make dynamic verification, invariant analysis or model-checking. This feature is not fully explored in the present work.

The transformation of analytical services in logical services represented by the model of service-oriented formal specification is supported by a set of processes based on MBSE, which attends the ISO/IEC 15288 (Haskins 2006) and are directed by OOSEM methodology (Friedenthal 2009), resulting in a model represented in SysML, called Service-Oriented Design Model.

The methodology OOSEM integrates a model-based approach and SysML to support the specification analysis design and verification of systems. OOSEM uses concepts of object orientation in conjunction with the methods of traditional systems engineering (top-down approach), supporting the specification of flexible and extensible service systems. OOSEM also intends to facilitate integration among object-oriented software development, hardware development, and testing.

In OOSEM organization a model is patterned through a SysML package structure. Thus, a Model Organization is a recursive structure representing the hierarchy of packages where a package is defined for each block, which in its turn is decomposed into packages, composing the next abstraction level. Figure 6 shows the package diagram named Model Organization containing the organization of other packages that make up the project presented in (Friedenthal 2009).

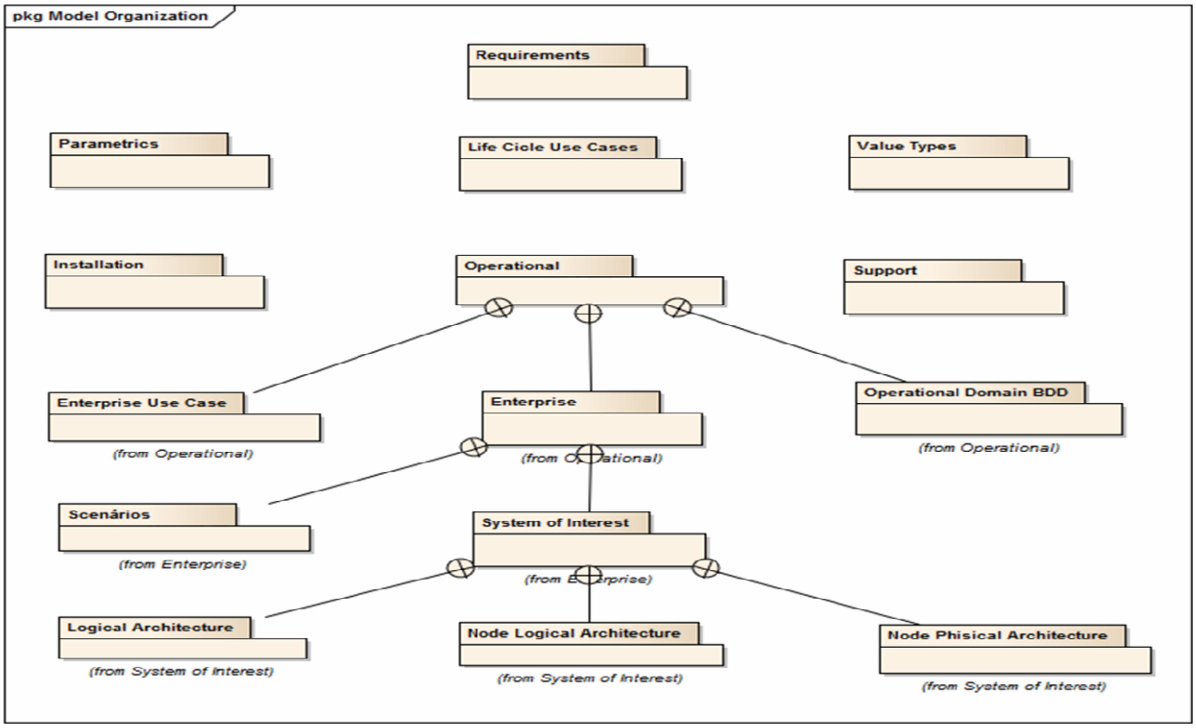

Figure 6. OOSEM Model organization (Friedenthal, 2009)

OOSEM uses a set of models in SysML to represent different artifacts generated by the activities of system development, which allow the capture, analysis and specification of service systems and its components in order to ensure consistency between the requirements of viewpoints. These artifacts can be refined and reused to support the evolution and maintenance of service systems.

Enterprise Architect, adopted here as a host framework, provides model verification, simulation capabilities and implementation of behavioral models formally represented in SysML. It also ensures that the proposed frame can be adapted to almost any available and consolidated platform, which put together theoretic approach and practical capabilities.

3.4 Formal Environment-ForEnv

ForEnv provides services associated with the evolution step from (basic and BP) requirements to formal specification and the preparation to the process of verification-which could eventually be done by model checking.

The last step of DesEnv translate a Service-Oriented Design Model from UML to SysML and transfer it to the Formal Environment using MDA representation offered by Enterprise Architect. The ManEnv acts in coordinates the transfer of models and give feedback on the results of this process.

Once the transference is completed, the model specification in SysML ForEnv should check for errors in the process and proceed to the formal verification of the dynamic behavior of the system. That could be achieved directly or by transforming all diagrams related to state-transition to GHENeSys network expressed in XML. The model can then be analyzed in the GHENeSys environment (a public domain tool) to check for deadlocks and to perform invariant analysis that could detect missing relations or some problems in the processes of the service system. Also a simulation can be performed in the GHENeSys environment.

The Petri net GHENeSys is obtained by applying the proposed method in Baresi & Pezze (2001), which deals with the transformation of the state diagram in Petri nets. Figure 14 shows the network GHENeSys resulting from the transformation of the state diagram of Figure 13.

The analysis of the properties of Petri net GHENeSys and their invariants (Figure 15) lets see if the behavior specified in the service design meets the requirements. The GHENeSys system also has a model checking tool which can operate with dense time or with time intervals, which can be very important to analyze time dependencies in service systems.

At this moment GHENeSys operates as an independent environment and the result of the analysis (timed or untimed) are not fully integrated in ManEnv. A file with the result of the analysis in GHENeSys is provided which is manually handled in the current stage of the framework. In the future a better integration is planned so that the ManEnv can open and analyze the file that comes from GHENeSys.

3.5 Managerial Environment-ManEnv

ManEnv offers management services activities and control the information flow SoftDiss, supporting planning and control activities, allocating resources and enabling the flow of information between different environments.

ManEnv works directly on maintaining the fundamentals of service orientation where resources are integrated, aiming customer-focused value creation which in its turn serves as an operant resource participating in the process of co-creation. The ManEnv deals with the maintenance of the ISS life cycle from identification to formal verification including also the requests for revision and improvements in ISS already in operation, resulting in a new specification or even the extinction of the ISS when they fail to fulfill its main function.

The ManEnv is supported by ISO/IEC 15288, specifically the processes planning, evaluation and control, complemented by processes for management risk, configuration and information.

The planning process establishes the scope, resources and schedule of tasks associated with the project supported by SoftDiss, setting the direction and infrastructure needed for the evaluation and project control. The assessment process collects and evaluates the progress of project by comparing the results with the planning, enabling monitoring and evaluation of project maturity. The control process uses evaluation to direct the efforts of the project, by directing resources to maintain project progress and address any deviations from the plan.

Risk management deals with the uncertainties in the lifecycle of the system. This process is used to understand and avoid potential impacts to cost, schedule and scope of the project, with a proactive and structured approach.

The objective of configuration management is to ensure effective management of the evolving configuration of a system, both hardware and software, during its life cycle. Information Management ensures that information is properly stored, maintained, validated, and accessible to those who need it thereby establishing/maintaining integrity of relevant system life cycle artifacts (Haskins 2006).

4. INFORMATION SYSTEM SERVICE-SMART GRID IN BRAZIL

Power Supply systems are very conservative all over the world wherever be the origin of the energy. In hydroelectric power plants in special, many decisions implementing the system in the last 120 years point to a treatment good-oriented, which also result in large transmission losses and waste in the costumer consumption. In fact, the electricity sector has remained virtually unchanged for nearly a hundred years, compared to other sectors such as telecommunications, computers and electronics.

Current power grid is not prepared to face modern challenges, and still faces many systemic problems, such as security threats to energy suppliers, including cyber attack, limited sources of alternative energy generation, lack of solutions to save energy, difficulty to ensure a constant and balanced supply of energy, as well as some problems to manage the distribution network. Thus, one of the greatest current technological challenges is to transform the current energy network in a dynamic, flexible and adaptable service system which could respond the requirements of a diversity of different classes of costumers.

A recent process, called Smart Grid, has been considered as a basis to update the energy sector, by incorporating of several technological advances, both in developed and emerging countries, including Brazil. The benefits obtained with Smart Grid can be significant, allowing consumers to, among other things, to get a finer control of energy consumption, integrating a new generation of hybrid systems, electric devices and appliances to combine different sources of power generation in order increase energy efficiency and achieve a better use of renewable sources.

According to EPRI (2008) the term Smart Grid is the overlap of a unified communications and control system on the existing infrastructure of the power grid to provide information to interested parties. It is a system that optimizes the distribution and delivery of energy, minimizing losses, makes adjustments and allowing a next generation of devices and applications focused on energy efficiency.

United States and Europe are working more intensively on the deployment of Smart Grid models. Europe is working on the use of distributed and renewable sources of energy, searching for emission reduction, energy efficiency and renewable generation.

U.S. intends to invest about U.S. $ 4.5 billion in Smart Grid, through a comprehensive program involving distribution companies, equipment manufacturers, telecommunications and IT companies along with government agencies such as DOE-U.S. Department of Energy and NIST-National Institute of Standards, in order to set standards for Smart Grid, interoperability to ensure protocols and equipment in many related areas, such as telecommunications, information technology and energy. Australia, China and Korea have similar reasons to reinforce the deployment of Smart Grid (NIST 2010).

Brazilian electricity system is unique in the world, strongly based on renewable hydrographic sources. The country-like US-has continental dimensions that makes integration, generation and transmission infrastructure based on hydroelectric power a challenge as well as its combination with distribution. It is certainly a different problem from those of Europe and U.S., which, in despite of its territorial size has different characteristics in the relation between distribution and generation, due to the use of thermoelectric power and the presence of nuclear power systems.

The questioning of the need for a flexible distribution network, accessible, reliable and economical, as observed in some locations in the U.S. and European countries, precedes the challenges and the implementation of Smart Grid in Brazil. However, Smart Grid is necessary and appropriate for Brazil, but with different motivations than the rest of the world. In Brazil there is a strong demand for contributions to improve the quality of service to the final consumer, while improving the reliability of the national electric system and reducing the waste.

According to the EIA-U.S. Energy Information Administration (http://www.eia.gov/ countries/data.cfm), Brazil has lost 17% of the energy produced, a result of extensive transmission lines between the hydroelectric generators and large urban centers, and also due to the amount of energy stolen in the urban centers, estimated at 8.7% of energy produced (IEA 2011).

4.1 Smart Grid Urban in Brazil

Thus, a good example of a service system is the case of urban Smart Grid, whose goal is to provide available energy (predominantly electrical) for individual, institutional and commercial consumers, in a sustainable manner, without waste, in accordance with their needs, avoiding deviations and losses. For simplicity, we confine the system to retail, that is, excluding users who demand more sophisticated services provided by substations.

Urban Smart Grid presents the electric utility as the basis of distribution, and clusters of processors which are made a load distribution and end users connected to a transformer. This architecture is relational and informative, that is, users can manage their consumption, sending restrictions, service interruptions or preferences, in order to provide sustainability, rational use of energy and avoid waste. The problem is also simplified by assuming that these users are just consumers, and for the current modeling cannot become small energy suppliers.

The information on the status of the transformers is used by an intermediate service to provide load balancing and information on the clusters and is synthesized by the concessionaire. Even with all these simplifications we have a system in three layers (a tree) with information flowing in both directions: from the leaves to the root and root to the leaves. The simplifications were made just to reduce a more sophisticated graph to a tree.

In fact this is a service system where the information flows while the energy is supplied and costumer can co-create value by saving energy in general or in rush hours while everybody contributes to reduce losses by stolen charge. The ultimate goal is service, that is, meet the different demands of end users, the most rational (from the viewpoint of the distributor) and sustainable (in terms of regulators). We therefore have an intricate system of viewpoints, which restricts the flow of energy and whose greatest achievement is through streamlined service from all viewpoints simultaneously.

The proposed treatment is precisely that of a service system, here translated into an information system coupled to a system of providing direct service. Note that the implement-tation of business is associated with the connection between the information system and the provision of energy, so it is almost impossible without the matching thereof.

Requirements management is crucial to the success of projects, especially when development environments and application are disjoint, i.e., users and stakeholders fall within an area with little or no intersection with the area of engineering, computer science and automation processes elicitation where requirements analysis become quite complicated. One way to alleviate the difficulty of dealing with disjoint domains is the use reference models or reference architectures, which allow a closer relationship between development environments and application.

The project IntelliGrid (http://intelligrid.epri.com/), originally called Project IECSA (Integrated Energy and Communications System Architecture), is sponsored by EPRI- Electric Power Research Institute with the goal of creating a technical basis for a Smart Grid. Main result was published in 2003, and the proposal of IntelliGrid Architecture (http://www.intelligrid.info/), were based on open standards, requirement-orientation and aimed at integrating data networks and equipment, while allowing interoperability between products and systems, providing the electric utility methodology, tools and recommend standards and technologies for system implementation. Intelligrid Architecture is used as a reference model, allowing an approach based on domain, close to processing requirements and specification of information systems service to generate a new urban Smart Grid.

4.2 SoftDiss applied to urban Smart Grid

SoftDiss is an environment for eliciting, documenting, analyzing and modeling requirements of ISS where the input is a request to solve a given problem and the output is the formal specification of the ISS.

ReqEnv offers services organization and record the various viewpoints of requirements to be considered in the specification of the ISS. The ReqEnv is supplied with information from Intelligrid Architect in order to specify an information system service designed to control the Digital Consumer Indicator-DIC, with the following functions:

- Request distribution network authentication: Authentication must request, wait for the answer and then store the authentication.
- Show the user the normal consumption, who should receive a message that will be treated for their identification, measurement data processing and display on screen the normal consumption.
- Allow user to request your instant power: the user can receive the instantaneous power (IP) by pressing a button and display IP on the LCD.
- Display posts from Center Supervisors for the end user: the messages are displayed on the LCD, the messages can not be displayed at the same time.

The ReqEnv uses reference model Intelligrid Architecture to capture two main viewpoints, consumer and energy service provider.

DesEnv offers services to identify and analyze the services necessary to solve the problem provided at the entrance of SoftDiss and result in formal specification of ISS, through the performance of three internal environments ConEnv, AnaEnv and LogEnv.

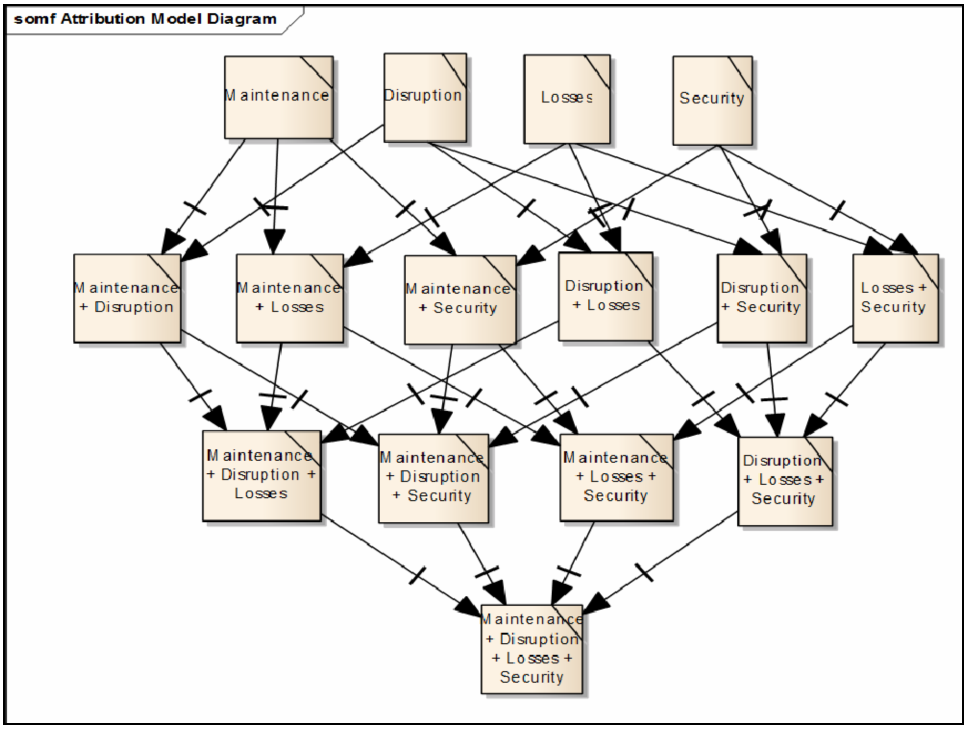

Figure 7. Attribution Model

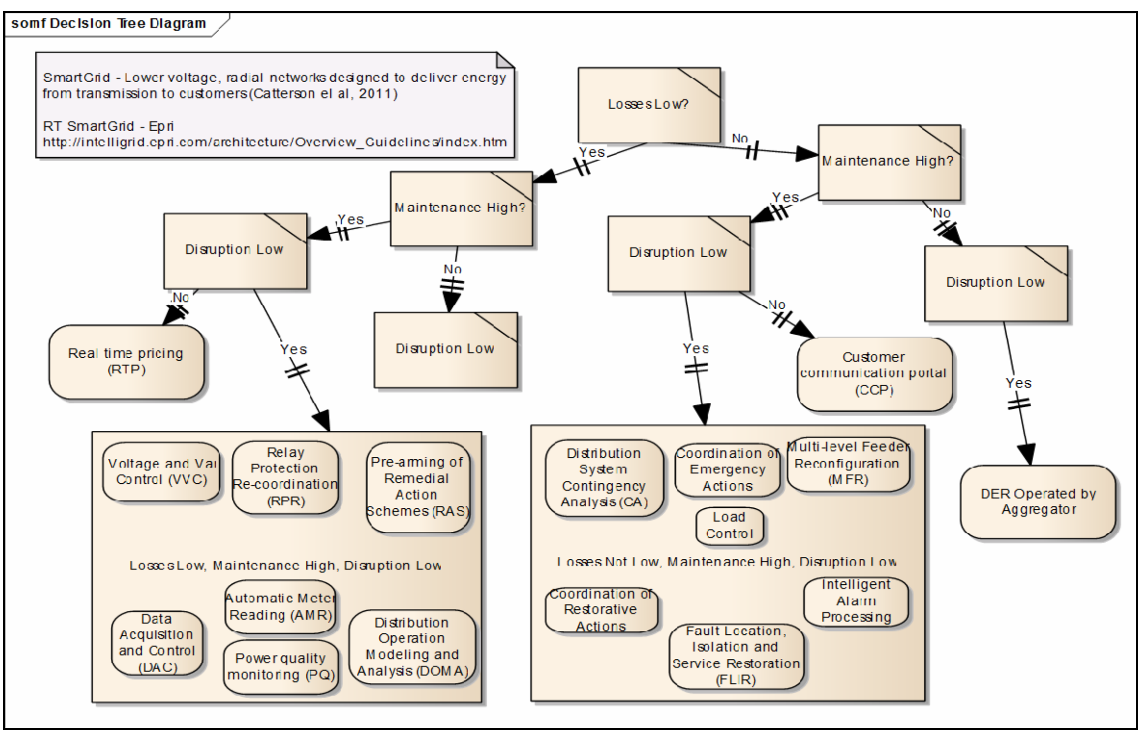

Figure 8. Conceptual Decision Tree

The ConEnv uses ReqEnv service and associated with a service system Smart Grid can determine, for example, the essential attributes: maintaining, power loss, loss and safety. The attribution model of Figure 7 is obtained from the combination of the essential attributes in the first level and the combination of attributes to the level immediately below, the latter being at the convergence of all attributes.

The model attributes provides information for the identification and discovery service, through the analysis of combinations obtained and the path in the network allows the determination of decision trees that guide the identification, discovery, and also provide a means of traceability between attributes and services which will be identified in the identification of conceptual services, as result, the conceptual decision tree in Figure 8, obtained from the essential attributes-losses, maintenance and disruption.

The decision tree has its root in the losses attribute and is associated with each level the decision to the attendance of other attributes, maintenance and disruption. The conceptual services identified-located in the leaves of the decision tree-are linked to attributes served by these services. The requirements for the discovery and identification of conceptual services represented in Figure 8 are obtained through the management exercised by the ManEnv, always considering the various viewpoints in ReqEnv.

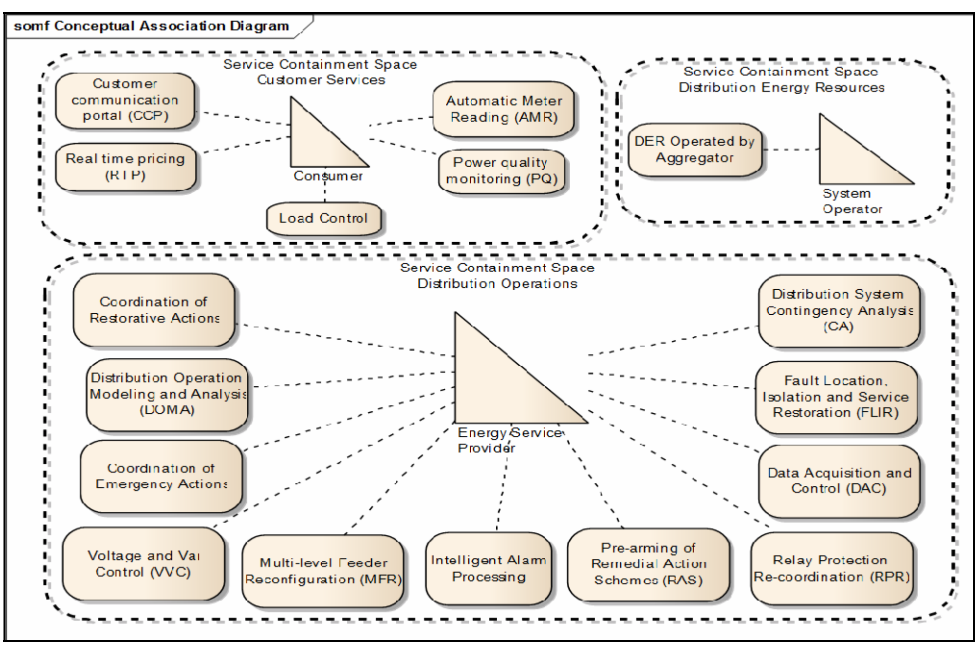

Figure 9. Conceptual Association Model

The conceptual services identified can be classified and grouped by the methods of conceptual association, organizing the services in different architectures: hierarchical, star, network and circular. Figure 9 shows a conceptual model of association for the services identified in Figure 8. The conceptual services of Figure 9 are connected to their respective service consumers.

In the SoftDiss AnaEnv, services RTP and AMR (Automatic Meter Reading) are connected by a coupling connector to the sub-services RTP and AMR which are expanded to their associated services through a connector atomic composition (Figure 10).

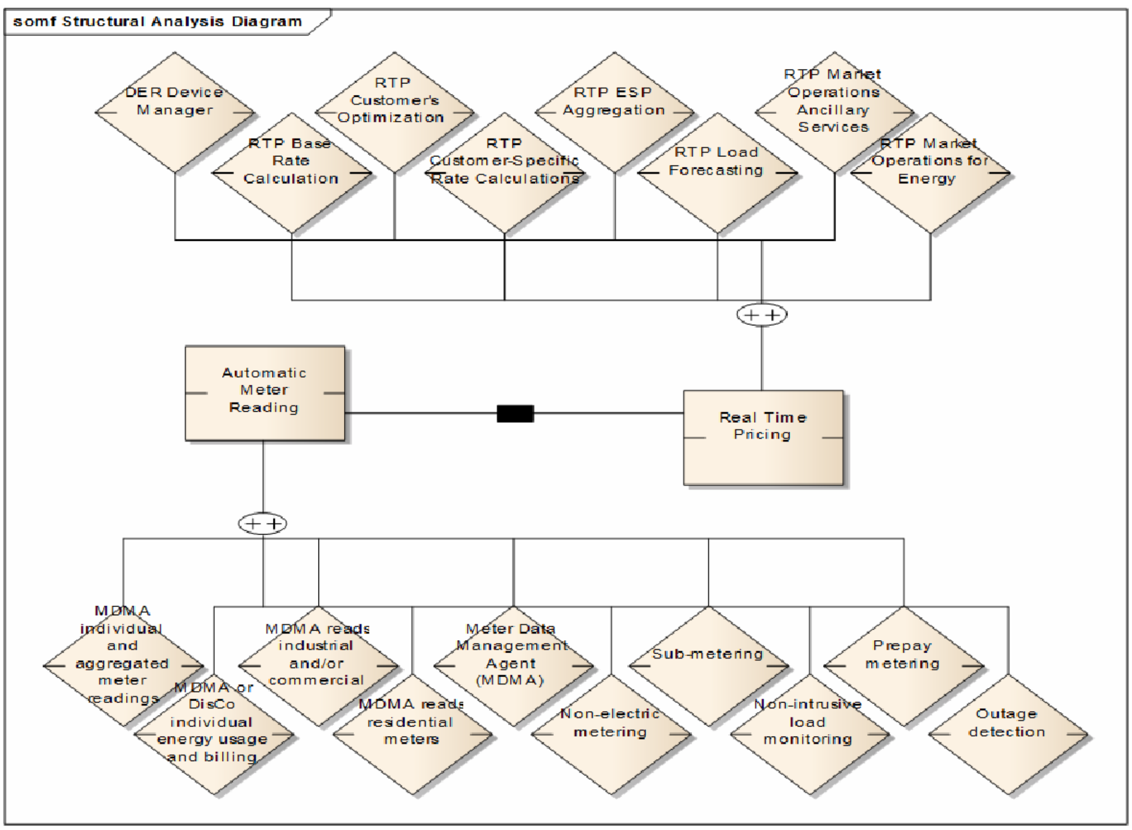

Figure 10. Structural Analysis Model

Figure 11 shows the business model integration of the services RTP and AMR. The perspectives of Operation and Maintenance are associated with the services ("integrated" connector) and with the business domains ("perspective of" connector)-Distribution Operation, Operation Market, Consumer Service and Distributed Resources.

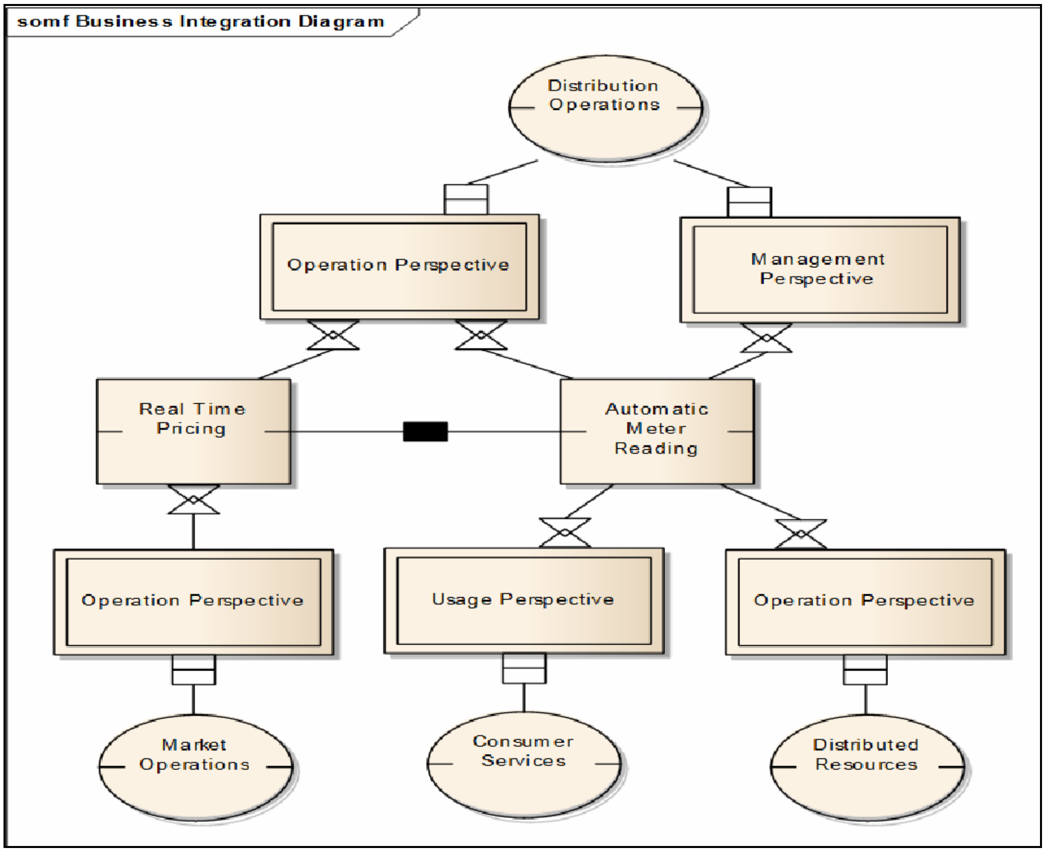

Figure 11. Business Integration Model

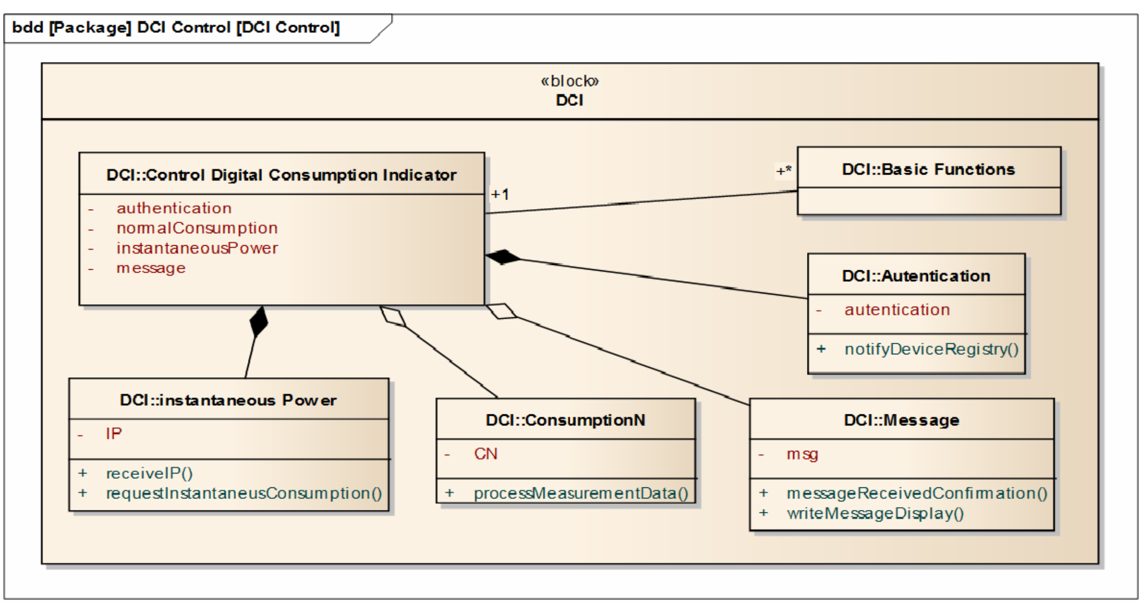

Figure 12. Logical model of relationship and service composition in SysML

LogEnv transforms analytical services in more tangible elements allowing depict relationships, structure and behavior of services enabling the creation of a formal architecture for services. In the case of Smart Grid urban part of this specification is shown in Figure 12 and Figure 13.

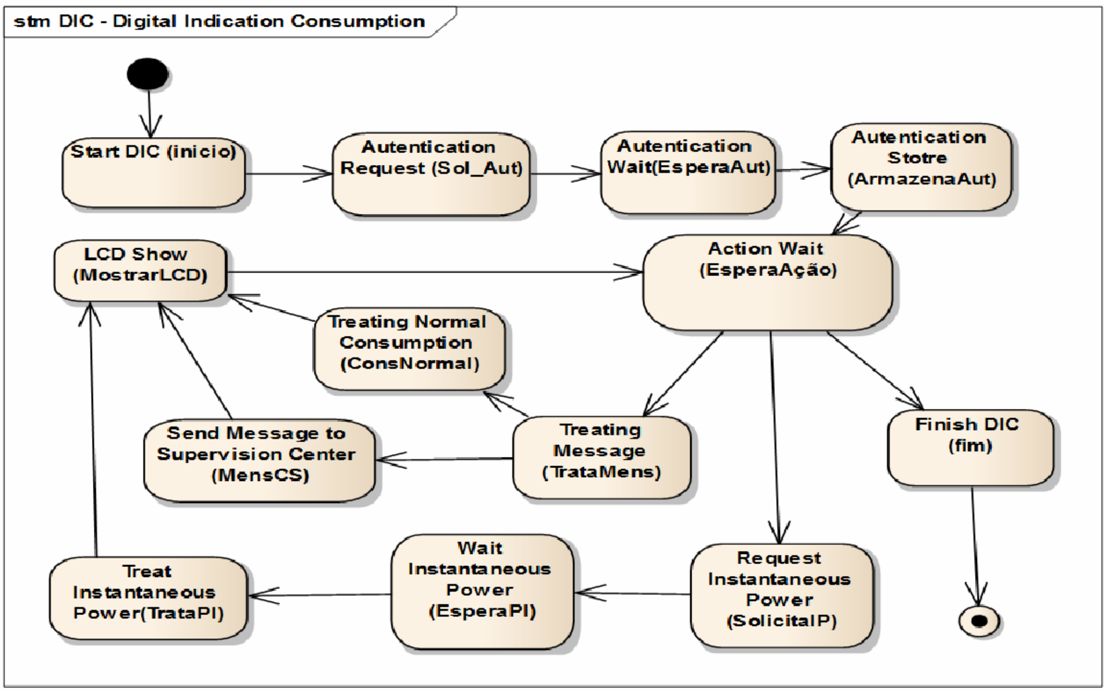

Figure 13. State machine model in SysML

ForEnv provides services associated with treatment adherence between the requirements of different viewpoints and representation of formal specification verification through simulation and formal mathematics.

The Petri net GHENeSys is obtained by applying the proposed method in Baresi and Pezze (2001), which deals with the transformation of the state diagram in Petri nets. Figure 14 shows the network GHENeSys resulting from the transformation of the state diagram of Figure 13. The invariants are calculated using the tool GHENeSys and shown in Figure 15.

The "place invariants" provide the equation:

M(Inicio)+M(SolAut)+M(EsperaResp)+M(TrataPI)+M(Fim)+M(EsperaAut)

$+M(ArmazenaAut)+M(EsperaAcao)+M(TrataMensagem)+M(ConsumoNormal)$

$+M(MostrarLCD)+M(MensagemCS)+M(SolicitaIP) = 1$

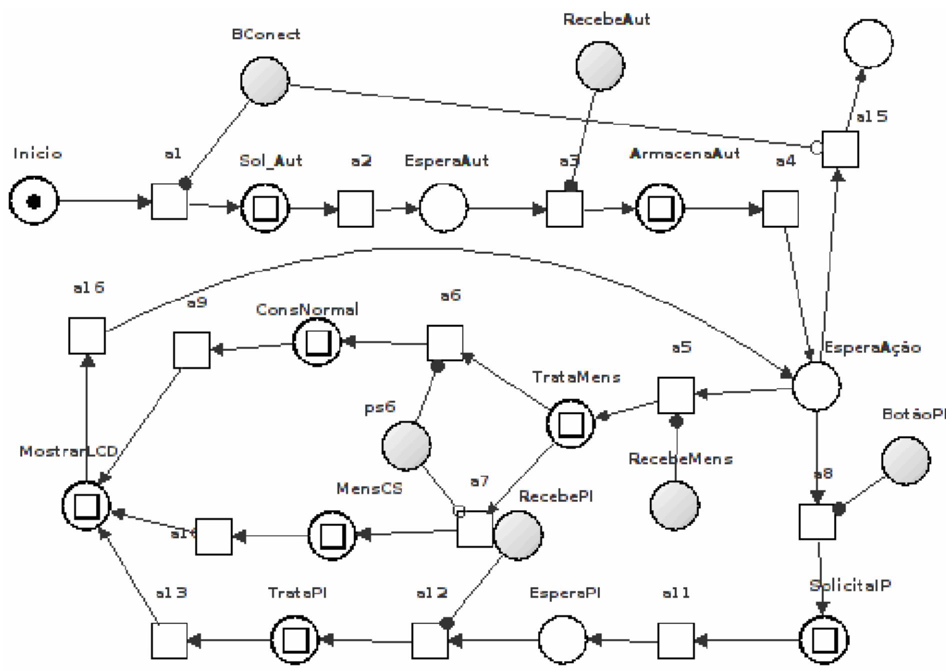

Figure 14. GHENeSys extended Petri Net for requirement verification

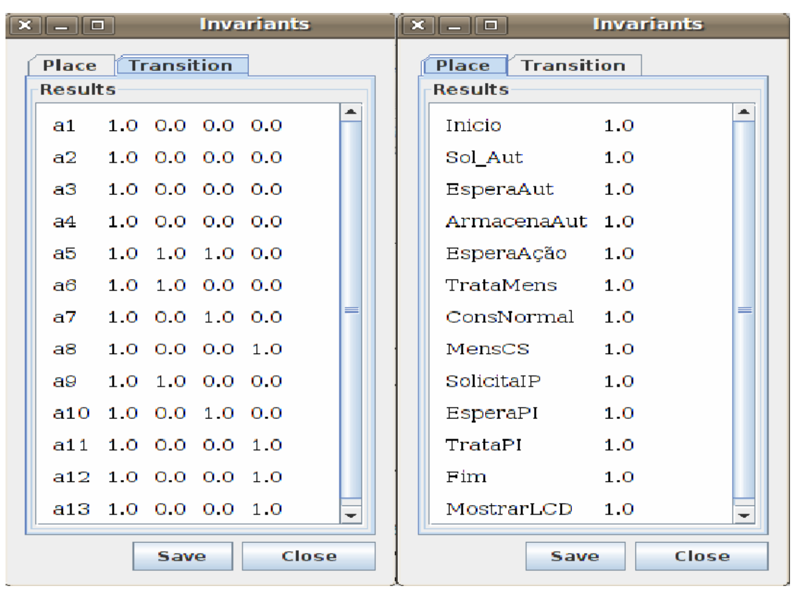

Figure 15. Transition invariants

By the analysis of place invariant equation it is concluded that the LCD display can only be one of the selected messages, which is the expected behavior. Considering the place invariant it is concluded that the transitions a1, a2, a3 and a4 are triggered at least once, which means that the authentication is performed when, given a fundamental requirement of DIC.

5. DISCUSSION AND FUTURE RESEARCH

The objective of the present article is first of all, to call the attention to the Service Engineering in the scope of the general studies about Science Service. This is motivated by the fact that a huge demand for service systems is approaching and it is possible that we soon would be in the same situation that occur in the past and was called "software crisis". New service systems are composed systems of systems, large in size and complexity, which design process demands a systemic and service-oriented approach. SoftDiss is a frame with these characteristics and was present as a concept proof. Among these important characteristics we detach: i) the model driven approach; ii) the adherence to current UML 2.0 tendency; iii) a system of system approach that include different new and consolidated systems available; iv) adherence to available commercial systems-even if a specific one had to be selected to drive implementation; v) a special treatment to the early phases and special attention to viewpoint analysis; vi) detailed attention to the matching between BP and basic requirements; vii) general tool do integration and management.

In SoftDiss proposal ally best practices and formal methods as any system which aim is to co-create value, in this case between human developers and the systems enrolled in the process of design (here played by Enterprise Architecture, GHENeSys, and the frame SOMF which was enhanced in some of its environments to fit the current proposal). However, all this process could not obfuscate the flexibility to be adapted to any other domain of tools and design disciplines that is not model driven engineering, object-oriented analysis and design.

It must be pointed that some disciplines could not be dispensed, such us traceability and viewpoint analysis in the requirement phases. Also the interaction among agents to validate requirements is a key issue in the new service engineering, since service systems must be a convergence of several and different expectations and needs from different class of agents, and able to generate value (or different value) with each one of these classes of agents. In new developments we would like to introduce agent intentions in this analysis, which point to i* requirement approach (Yu 1997).

Another missing point is that new service systems must be collaborative and therefore intelligent. This would affect the target system itself, result of the design process, but would also affect the design process.

Most demanding environment to intelligent processes is the ManEnv. Two processes of the ManEnv are particularly demanding: the treatment of requirements-especially when an arrangement of basic services to not fit completely the general requirements, or when ManEnv need to plan actions and interventions that would show very helpful in the process. We do not include this here because this was not implemented and even the best techniques to use are still an open discussion in DesignLab. Some attempts have being made and it is possible that in the near future we try to include another available tool: itSIMPLE (Vaquero and Silva and Beck 2011), which were also developed by DesignLab. Even if the whole system is not added some of the algorithms developed there could be migrated to SoftDiss.

Finally, service engineering is very dependent on communication since the general idea is that the service system be always collaborative and fully integrated with its actors. That could be extended to the design process itself where there is another missing point in the proposal presented here: reusability.

However, to get an efficient use of reuse we must face a problem that different projects use a different set of words and terminology, especially if there is a difference in the domain they are supposed to serve. In the past, DesignLab worked in this subject by using metaphors (or a formal theory of metaphors) to find similarities of derivation of terms in different set of domain discourse. In the near future that will be also an interesting way to face communication problems in the design process of service systems. The validation process of requirements could also benefit from this feature since the same problem can occur between the discourses of different classes of agents.

Whatever be the research agenda for the future it is urgent to start the discussion about the composition of techniques in service engineering, not only because the scientific appeal but for practical reasons as well.

ACKNOWLEDGMENTS

The authors acknowledge Sparx Systems for the king agreement that make it easier to use the Enterprise Architecture in this and other research projects of DesignLab. We also acknowledge DesignLab team that contributes in several points of the current work and also Prof. Raimundo Gomes and Prof. José Rubens Sicchar for the contribution in the Smart Grid project which implementation is now in Amazon State University.

REFERENCES

Aalst W (1998) The Aplication of petri nets to workflow management. The Journal of Circuits, Systems and Computers 8(1):1-53.

Aalst W & Hee V (2002) Workflow management: models, methods and systems. Cambridge: MIT Press, 2002.

Bardhan IR, Demirkan H, Kannan PK, Kauffman RJ, and Sougstad R (2010) An Interdisciplinary Perspective on IT Services Management and Service Science. Journal of Management Information Systems 26(4):13-64.

Baresi L & Pezzè M (2001) Improving UML with Petri Nets. Electronic Notes in Theoretical Computer Science 44(4):107-119.

Barile S & Polese F (2010) Smart Service Systems and Viable Service Systems: Applying Systems Theory to Service Science. Service Science Journal 2(1/2):21-40.

Beer S (1984) The Viable System Model: Its Provenance, Development, Methodology and Pathology. The Journal of the Operational Research Society 35(1):7-25.

Bell M (2008) Service-Oriented Modeling: Service Analysis, Design, and Architecture. 1st ed. New Jersey: John Wiley.

Bertalanffy L (1950) An outline of general system theory The British Journal for the Philosophy of science 1(2):134.

Boulding KE (1956) General systems theory-the skeleton of science. Management science 6 (1):127-139.

Cambridge (2007) Succeeding through service innovation. Symposium A Quarterly Journal In Modern Foreign Literatures.

Carr JJ (2000) Requirements Engineering and Management: The Key to Designing Quality

Complex Systems. The TQM Magazine 12(6):400-407.

Chesbrough H and Spohrer JIM (2006) A Research Manifesto for Services Science. Communications of the ACM 7(49):35-40.

Davenport T (1993) Process Innovation: Reengineering Work Through Information Technology. Boston: Harvard Business School Press.

Desel J & Reisig W (1998) Place/transition Petri Nets. Lectures on Petri Nets I: Basic Models:122-173.

EPRI (2008) The Green Grid-Energy Savings and Carbon Emissions Reductions Enabled by a Smart Grid. Manager. Palo alto, CA.

Estefan JA (2007) Survey of Model-based Systems Engineering (MBSE) Methodologies. Incose MBSE Focus Group.

Friedenthal S, Moore A, & Steiner R (2009) A practical guide to SysML: the systems modeling language, San Francisco: Morgan Kaufmann.

Gotel OCZ & Finkelstein CW (1994) An Analysis of the Requirements Traceability Problem. In Proceedings of 1st International Conference on Requirements Engineering, 94-101. IEEE Computer Society Press.

Haskins C (2006) Systems Engineering Handbook. INCOSE.

Hillah L, Kordon F, & Petrucci L (2006) PN standardisation: a survey. Lecture Notes in Computer Science: Formal Techniques for Networked and Distributed Systems-FORTE: 307-322.

IEA (2011) Key World Energy Statistics 2011. Statistics. OECD Publishing. www.iea.org.

Jensen K (1998) An Introduction to the Practical Use of Coloured Petri Nets. Lectures Notes in Computer Science:237-292.

Kent S (2002) Model Driven Engineering. In Integrated Formal Methods 1:286-298. Springer.

Khadraoui A & Feltus C (2012) Service Specification and Service Compliance: How to Consider the Responsibility Dimension? Journal of Service Science Research 4(1):123-142.

Kim, YJ & Nam K (2009) Service Systems and Service Innovation: Toward the Theory of Service Systems. In AMCIS Proceedings.

Kobryn C (2004) UML 3.0 and the Future of Modeling. Software and Systems Modeling:4-8.

Kotonya G & Sommerville I (1998) Requirements Engineering: Processes and Techniques. John Wiley and Sons Ltd.

Leffingwell D & Widrig D (2000) Managing Software Requirements: a Unified Approach. New Jersey: Addison-Wesley.

Leite J & Prado CSD (1996) Viewpoints on Viewpoints In ACM Workshop SIGSOFT:285-288. ACM Press.

Letelier P (2002) A Framework for Requirements Traceability in UML-based Projects. In 1st International Workshop on Traceability in Emerging Forms of Software Engineering. In Conjunction with the 17th IEEE International Conference on Automated Software Engineering.

Li H-F, Wang J-F, Yu H-L, & Yang D-L (2007) Services Science, Management, and Engineering: A Literature Review in the Perspective of Management Science. In Proceding of the IEEE IEEM:1438-1441.

Lusch R, Vargo S, and Wessels G (2008) Towards a Conceptual Foundation for Service Science: Contributions from Service-dominant Logic. IBM Systems Journal:1-9.

Maglio PP, Vargo SL, Caswell N, & Spohrer J (2009) The Service System Is the Basic Abstraction of Service Science. Information Systems and e-Business Management 7(4): 395-406.

Marshall C (2000) Enterprise Modeling with UML. Addison-Wesley.

Miyagi P (1988) Control System Design, Analysis and Implementation of Discrete Event Production Systems by using Mark Flow Graph. Ph.D. thesis, Tokyo Institute of Technology, Tokyo.

Moussa S & Touzani M (2010) A Literature Review of Service Research Since 1993. Journal of Service Science 2(2):173-212.

Murata T (2002) Petri Nets: Properties, Analysis and Applications. Proceedings of the IEEE 77(4):541-580.

Ng ICL & Maull R (2009) Embedding the New Discipline of Service Science. In The Science of Service Systems, ed. Demirkan, H, J.H. Spohrer and V. Krishna Eds. Springer. 44.

NIST (2010) NIST Framework and Roadmap for Smart Grid Interoperability Standards. Nist Special Publication.

OMGSysML (2012) OMG Systems Modeling Language (OMG SysML V1.3). http://www.omg.org/spec/SysML/1.3/.

Oliveira VC & Silva JR (2011) Process Oriented Design Service: Case Study for Automated Information System. In 21st Brazilian Congress of Mechanical Engineering, Natal, RN, Brasil 10.

Ostrom AL, Bitner MJ, Brown SW, Burkhard KA, Goul M, Smith-Daniels V., Demirkan H, & Rabinovich E (2010) Moving Forward and Making a Difference: Research Priorities for the Science of Service. Journal of Service Research 13(1):4-36.

Pawson R (2002) Naked Objects. Software, IEEE:81-83.

Pearce P & Hause M (2012) Model-Based Submarine Design. In SETE and APCOSE. Brisbane, Australia, 15.

Raumbaugh JR, Blaha MR, Lorensen W, Eddy F, and Premerlani W (1991) Object-Oriented Modeling and Design. Prentice-hall.

Robertson S & Robertson J (2006) Mastering the Requirements Process. Second Edi. Addison-Wesley.

Silva JR, Benítez I, Villafruela L, Gomis O, & Sudrià A (2008) Modeling Extended Petri Nets Compatible with GHENeSys IEC61131 for Industrial Automation. The International Journal of Advanced Manufacturing Technology 36(11):1180-1190.

Silva JR & Miyagi PE (1995) PFS/MFG: A High Level Net for the Modeling of Discrete Manufacturing Systems. In Balanced Automation Systems, Arquitectures and Design Methods, ed. Luis Camarinha Matos and Hamideh Afsarmanesh, Chapman and Hall: 349-362

Silva JR & Miyagi PE (1996) A Formal Approach to PFS/MFG: a Petri Net Representation of Discrete Manufacturing Systems. In Studies in Informatics and Control. Romenia: IC Publications.

Smith E (1998) Principles of High-level Net Theory. Lectures on Petri Nets I: Basic Models.

Sommerville I, Sawyer P, & Viller S (1997) Viewpoints for Requirements Elicitation: a Practical Approach. In Proceedings of IEEE International Symposium on Requirements Engineering: RE 1998. IEEE Comput. Soc:74-81.

Spohrer J, Maglio PP, Bailey J, & Gruhl D (2007) Steps Toward a Service Systems. IEEE

Computer Society:71-77.

Spohrer J, Vargo SL, Caswell N, & Maglio PP (2008) The Service System Is the Basic Abstraction of Service Science. Proceedings of the 41st Annual Hawaii International Conference on System Sciences (HICSS 2008):104-104.

Stair RM & Reynolds GW (2010) Information Systems. Australia: Cengage Learning.

Stanicek Z & Winkler M (2010) Service Systems Through The Prism of Conceptual Modeling. Service Science Journal 2:112-125.

Tang A, Jin Y, & Han J (2007) A Rationale-based Architecture Model for Design Traceability and Reasoning. Journal of Systems and Software 80(6):918-934.

Vaquero T, Silva J, & Beck J (2011) A Brief Review of Tools and Methods for Knowledge Engineering for Planning and Scheduling. KEPS-ICAPS. Freiburg. Germany.

Vargo SL & Akaka MA (2009) Service-Dominant Logic as a Foundation for Service Science: Clarifications. Service Science Journal 1(1):32-41.

Yu ESK (1997) Towards Modelling and Reasoning Support for Early-phase Requirements Engineering. Proceedings of ISRE 3rd IEEE International Symposium on Requirements Engineering:226-235.

Zhao G & Perros H (2009) How service science management and engineering (SSME) can evolve to an academic discipline?. International Journal of Web Engineering 5(4).

AUTHOR BIOGRAPHIES

Valter Castelhano de Oliveira (Ph.D. in Engineering, Escola Politecnica of University of São Paulo, Brazil) is an undergraduate course coordinator at the State College of Technology of São Paulo. He has MSc degrees in Engineering from Escola Politecnica of University of São Paulo (POLI-USP) and in System Information Management from Pontificia Universidade Católica de Campinas. He also has a Bachelor degree in Computing Science from Universidade Federal de São Carlos. He has PMI-PMP with 25 years of experience (IT and software development). His current research interests are in service systems design and information systems.

José Reinaldo Silva (Ph.D. in Electric and Computer Engineering, Escola Politécnica da USP, Brazil) is an Associate Professor since 2006 in the Mechatronics Department at Universidade de São Paulo. He has a Bachelor degree in Physics and an MSc degree in Atomic Physics, both in Brazilian federal universities (UFBA and UFPE), and an MA degree in Interdisciplinary Computer Science from Mills College in USA. He had a postdoc stage in the Cheriton School of Computer Science and in the Systems Design Engineering Dept. at the University of Waterloo, Canada. His research interests are in engineering design including design methodology, requirements analysis, formal methods for modeling, verification and model checking using (Timed) Petri nets, intelligent systems including artificial planning and scheduling, information systems and service systems.