Issues and Challenges in Dynamic Systems Design and Engineering – A Value-Oriented Approach

João Pombinho[1], José Tribolet[1,2],

[1]CODE - Center for Organizational Design & Engineering, INOV, Rua Alves Redol 9, Lisbon, Portugal
[2]Department of Information Systems and Computer Science, Instituto Superior Técnico Technical University of Lisbon, Portugal
jpombinho@acm.org, jose.tribolet@inesc.pt

Abstract. Modeling organizations as complex systems in permanent evolution, as an answer to change dynamics, is an increasing challenge. Particularly, there is a lack of an integrated perspective that is generally and recursively applicable to organization chains, organizations and sub-organizations of several types and sizes. Our research aims to answer how to incorporate purpose into system development activities, in a way that promotes value-orientation and innovation. Three main conceptual challenges were identified: 1) the lack of capacity to view a system, and the services it provides, integrated in different value chains; 2) the separation of the instance of a system from the purpose behind its design; and 3) the conceptual unidirectionality of the system development process. In this paper, we present the proposal of rationalizing system design and engineering decisions with value-orientation, materialized in a set of principles and a four-layer framework: System, Service, Market (Value) and Problem Solving (Purpose).

Keywords: System Design and Engineering; Demand Management; Purpose; Market; Value Chains; Service Oriented Architectures.

1 Introduction and Motivation

The main premise of Organizational Engineering is that organizations are systems and, therefore, can be object of engineering activities. ICT-based organizations are especially promising candidates for this kind of approach since their processes are mostly immaterial, ranging from a few activities to the whole chain and even the final product or service. Events are generated and handled in ways that facilitate their capturing in comparison to other systems without explicit state representation. But regardless of the main type of agents that support the organization activities, the focus should turn from *doing things right* to *doing the right things*, as it is inglorious to have outstanding performance at something that should not be done at all. This is especially critical in ICT-based organizations, where the high level of automation allows for transactions to be executed massively. This fact amplifies any flaws in the creation process of such systems, which end up embodying requirements that are implemented without being formally aligned into an overall model. Even if the initial

implementation serves the purposes it was created for, the evaluation of impacts, conception and implementation of subsequent changes is difficult to perform in a rational manner due to modelling shortcomings.

In order to tackle these classical issues from an innovative perspective, we begin by asserting that every kind of organization, regardless of their composition and objectives (private or public, political, business, education, healthcare, non-profit, etc.) brings about some form of value, directly or indirectly, so this is a unifying concept. Also, a given system is *one possible solution (out of many) to a problem*; a *means*, not an *end*. The market does not request an organization; instead, it values the services that it provides and that contributes to a solution for a given problem. Therefore, it is the organization that should reconfigure itself as a system to have the capacity of providing the services requested by the market, not the other way around.

Formal organizations are generally created as providers of a *repeatable* and *stable* solution to a demand, meaning there is reasonable belief that its elements will be continuously available. The rationale behind this quest for stability is, essentially, the lack of agility in procuring resources on-demand, compromising between evaluating every possible solution to each business activity and the time and effort consumed in doing so. However, with the current change pace, stability is a luxury unavailable to most organizations as the demand set itself changes. Therefore, a framework must explicitly include the concept of market, with demand/offer dynamics.

In addition, being market-aware means *recognizing the user's freedom of choice* – in the end, in every chain there will always be an end-user! Even in operational dynamics, it is frequent that people use alternative, unofficial means of performing actions; not recognizing it as a choice, in a formal or informal market, is missing the opportunity to improve organizational design and engineering.

This paper reflects ongoing research and is structured as follows: Section 2 presents problem analysis with a motivation example from a Library DEMO model, which is the base for identifying current challenges. These are grouped in five problem areas, with the corresponding research questions and a brief and localized related work review. In Section 3, we present a set of principles currently applied in a real-world setting to tackle the identified issues, along with a Framework overview. The paper closes with conclusions and contribution summary in Section 4.

2 Problem Analysis

2.1 Base Theory: Systemics, DEMO and the GSDP

This paper addresses system development from a problem-solving perspective driven by value. The *system* definition we will use, from [1], defines the following properties for a system: *composition* – a set of elements of some category; *environment* – a set of elements of the same category, disjoint from the composition; *production* – things produced by elements in the composition and delivered to the environment; and *structure* – a set of influence bonds among the elements in the composition, and between them and the elements in the environment.

Design and Engineering Methodology for Organizations (DEMO) [1] is a cross-disciplinary theory for describing and explaining the structure and action of organizations. It defines an organization as a discrete dynamic system consisting of social actors, who enter to and are responsible for commitments with each other in a coordinated manner. Enterprise ontology is a model of an organization in which these commitments serve as models for business transactions. DEMO was chosen because it models the essence of transactions between responsible actors and abstracts away implementation issues. However, it is currently not widespread in terms of awareness by the community. Included in its theory set is the *Generic System Development Process* (GSDP), shown in Fig. 1, which begins with the need by a system, the *using system* (US), of a supporting system, called the *object system* (OS).

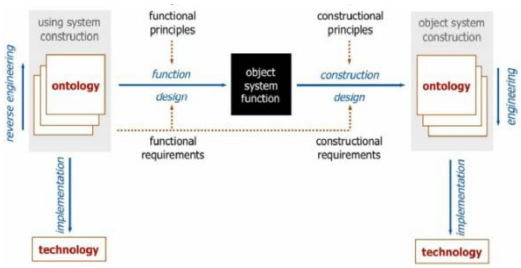

Fig. 1. Generic System Development Process [1].

From the white-box (WB) model of the US, one determines the functional requirements for the OS (function design), formulated in terms of the construction and operation of the US. Next, specifications for the construction and operation of the OS are devised, in terms of a WB model (construction design). The US may also provide constructional (non-functional) requirements. Choices are then made with each transition from the top-level white-box model towards the implementation model.

The GSDP has articulate and clear primitive concepts that reflect the essence of system development. We chose to use it as a reference, since we believe the critical analysis is extensible to other system development processes.

To close this brief presentation of the base theory set, it is important to differentiate two aspects of a system: *Teleological*, concerning its function and behaviour, a black-box; and *Ontological*, about its construction and operation, a white-box [1]. The main question is: *How to integrate the teleological and ontological conceptions of a system so that proactive innovation and value-orientation is promoted?*

2.2 Current Challenges Identification and Analysis

In order to clarify the problem space, constituted by a large set of core concepts from different concern areas, a practical scenario based on the classical DEMO Library case [1] will be used for instantiation. In this example, the elements of the system dealing with the membership (solid black line-bounded area in Fig. 2) are not justifiable as bringing direct value to the customer, who only wants to get hold of a

book. However, as it can be seen in Fig. 2, this is all but clear in the ontological (construction) model:

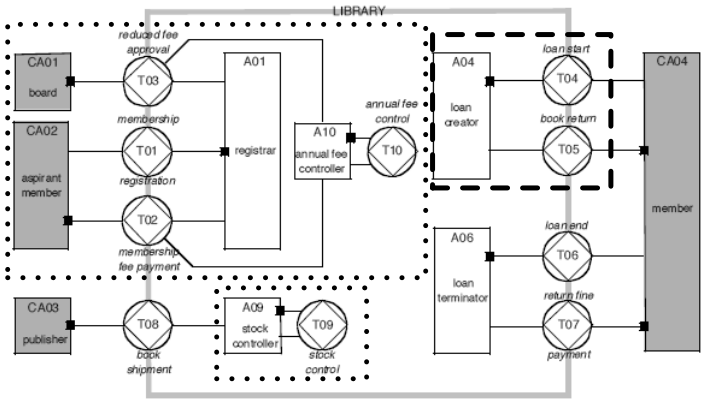

Fig. 2. Library example – Construction analysis.

Regarding the core business of *providing reading content*: 1) the *core service* is concealed in the area marked by a dashed line, obscured inside a *loan transaction*; 2) inside the solid black line, a *sacrifice* of the customer in obtaining the service and its support (sub)system; finally, the area bounded by points encloses a support process that may need revision, for instance, in a change scenario of going digital.

About the Membership Management subsystem, one must ask if there is really a customer who *wants* a membership or was this subsystem included in the Library as the manifestation of a strategy to get a fixed amount of income to face, for instance, stocking management? Is this still a problem if the organization does not pay for the books and space? Is it done for profit or simply as a response to the cost of keeping a large library? Is it part of the Library concept, i.e., every library also offers it by definition? Under what conditions should this **decision** be **reviewed**?

There are a number of approaches of different nature to parts of these problems, including system development by Dietz and Hoogervorst [1], Service Design by Bell [2], Enterprise Architecture by Lankhorst [3], Goal-orientation [4, 5] and Value Management by Gordijn [6] [7], to name a few; however, none of the questions can be answered directly by these or any other framework that we are aware of.

By analyzing the current State of the Art, the following five problem areas were isolated, with their respective Research Questions:

2.2.1 Value Definition

Value is, by nature, dependent on the stakeholder and, thus, relative. The problems in adequately naming and scoping of a service, know in the Service Design community, are a symptom of this [2]. Regarding the Library's purpose, what is the core transaction for providing value? For instance, should the transaction be named "Loan book" or "Provide (limited-time) access to (reading) content"? Is the "Membership registration" service is interesting *per se*, or is it only in the way of getting a book, that is specific to this particular construction of a library? This is why current goal-

oriented modeling [4, 5] is not enough: it lacks an independent value structure to refer to. It must be understood that this structure is not subordinate to the service-providing systems, but the other way around! e3Value [7] provides essential value mapping perspective but lacks a holistic and formal framework for enterprise modelling.

RQ 1: *How to 1) represent **value** as a manifestation of **purpose**, in a structured yet relative way and 2) **trace** it through system development deliverables?*

2.2.2 Value Production Semantics and Business Model Definition

Systems design and engineering activities are guided by principles and requirements, normally based on informal specifications such as textual descriptions of use cases. A system's production is the best alignment beacon as it is the effective contribution to its environment. Current approaches do not model the system's production in a way that can be engineered. In our example, the same construction would serve both a Book and a Music Library; is the loan mechanism an interesting way to provide both types of content? Also, what is the threshold where an organization ceases to be of a certain type and what are the more general and specific organization types? Business Model Canvas [8] is an interesting and pragmatic approach that shares this concern area but lacks the formality that allows effectively entering the system engineering phase. For instance, a Library without a Membership subsystem is still a Library; but is it still a Library without a Catalogue?

RQ 2: *How to represent the **semantics of a system's production** in a relevant way and how does it contribute to the **essential definition** of a system?*

2.2.3 System/Sub-system (De)Construction Modeling Support

The construction of a system resulting from the development process is a compiled structure that obscures the system/subsystem relations and their motivation. It is very hard to separate a given subsystem from its owner system, especially if it was modeled from a flat description of the operation of the organization, instead of a sequential bootstrap or an incremental design step.

Assuming the stability of a value chain is generally unsafe because of change dynamics, which justify the need for a structure where to represent multiple scenarios in order to provide a flexibility point instead of a frozen solution path. How does a Library compare to a Bookstore or a Publisher, from the customer's perspective?

RQ 3: *What concepts are needed to define **system and subsystem relations** so that they can be applied **recursively**? How to represent **multiple scenarios** regarding different solutions to a given problem in a flexible way?*

2.2.4 Lack of System Intervention Rationale Modeling

It is quite common that questions about system intervention rationale are very hard to answer, especially some time after it has happened. For instance, regarding the

introduction of the Membership subsystem: 1) When was the decision taken? 2) What was its purpose? Was it for mitigating the risk of non-return? 3) What were the design principles, constructional principles, assumptions and constraints applied? Are they still valid? For any kind of content the library may want to provide, e.g., e-books?

DEMO has been extended [9] to incorporate change dynamics but, at this time, still does not model the formal rationale of each change. This is particularly relevant in creating new, innovative, components of the organization, both in bootstrap and in on-going phases. The GSDP also does not prescribe what to do with the objects supporting the rationale of the decisions made during the process. The implementation steps consist in introducing restrictions on the construction, for instance: 1) assumptions, such as assuming the customer is necessarily a reader; 2) constraints, such as available technology to offer books, e.g., physical or digital.

RQ 4: *How to define the **rationale of a decision** in terms of the application of design principles, constructional principles, assumptions and constraints in a **structured** way that is relevant and explicitly include it in the system model?*

2.2.5 Conceptual Unidirectionality of the System Development Process

The unidirectionality of the system development process induces an upper limitation of the solution's value, indexed to the original functional request scope. Extra value that could be derived in bottom-up fashion, either available at the original design time or in future interventions, is not addressed. According to the GSDP, *Determining Requirements* is defined as 'The design phase that starts from the ontological model of the using system, and ends with the functional model of the object system' [1]. This approach requires full knowledge about the US, which is a serious limitation. Even if it were trivial, the solution would be irrecoverably restricted to satisfying the demand of a specific US, its value is limited from the outside instead of being allowed to expand creatively inside out. This is why the Agile [10] paradigm does not fully solve this issue, regardless of the length or frequency of the development cycles.

Again using the Library case, if e-books begin to be provided by the Library system, what are the possible USs for that new OS? For instance, a Printing on Demand (PoD) service requires no stock control of physical books.

RQ 5: *What is the process of supporting **innovation** regarding the essential definition of a system, and which concepts result from the introduction of **bidirectionality** in the **GDSP**?*

3 Towards a Solution: Principles and Framework Overview

3.1 Principles of a Different Way of Thinking

In this research, we are proposing a set of principles that were derived from practical application at a real-world Demand Management scenario:

- Recognize the system being developed as **one of many possible solutions** for a problem and, therefore, as a **means**, not an end;
- Conceptually integrate the **Teleological** and **Ontological** perspectives of a system by introducing the problem/solution paradigm and value concepts into system modeling activities;
- Improve **problem definition and elicitation** by using the concepts of system value, subsystem value generation and positioning the system in a demand/offer relation between consecutive nodes in a **value chain**;
- **Improve the clarity of system models**, by embedding value-semantics in the development process and **tracing** it to the relevant system elements as a structured means of expressing **purpose**;
- Look beyond the boundaries of formal organizations, into **value nets**, as a provider may serve multiple customers (n-1) with different problems and expectations, assisted by multiple suppliers (n+1) to increase design abstraction so that system value is increased as a result of greater market;
- Improve **change evaluation and decision rationale** by applying design principles, constructional principles, assumptions and constraints in a relevant, structured way that is explicitly included in the resulting model;
- Support **Innovation** by using these intermediate constructs from the development process to conceptually **reverse the development process** in a rational way during a **reengineering effort – Reverse Discovery.**

This set of principles reflects the current thinking and results, and in the course of research will be further refined and validated. It is important to note that it does not imply a specific way of working and is independent of tool support - even though it can be greatly aided by it, especially according to portfolio size and change rate.

3.2 Framework Overview

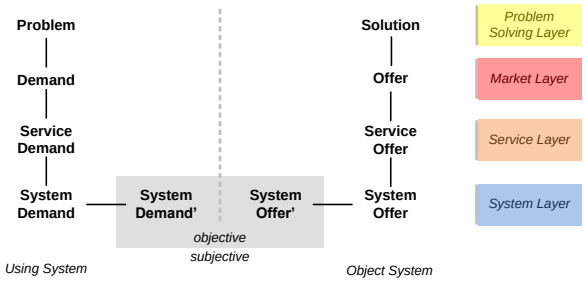

Fig. 3. Framework Overview

Our high-level solution proposal includes a four-layer framework: System, Service, Market (Value) and Problem Solving (Purpose); their relative positioning is represented in Fig. 3. The most differentiating concepts for each of these layers are presented next, in bottom-up fashion: from *system* towards *purpose*.

3.2.1 System Layer

For the purposes of this paper, we will focus on the *recursivity* property of our system definition. We argue that any given complex system can be decomposed into more granular systems chained together; the rationale for forming each link is the same that should exist between the components of a system for, in the end, the same concepts will recursively apply. We base the last statement in the following assertion:

If a single element is part of a system's *composition*, then it is connected by means of the system's *structure* to other elements; therefore, this connection must represent (but does not necessarily specify) the element's contribution to the *production*.

A single element of a system is also a system (a sub-system of the original system), with a *composition* constituted by a single element, an *environment* formed by the other elements in the original system, a *structure* linking the element to the environment and a *production* as the fact pertaining the *contribution* it makes to the production of the original system – which is its *purpose*, regarding that chain.

3.2.2 Service and Market Layers

These two layers are responsible for mediating the relation between a customer and the systems that participate in solving his problems. The service layer abstracts functionality from a given conceptual system in terms of inputs and outcomes while framing it in transactional semantics, with exchange of contract and operation conditions. In turn, the market layer uses value as a driver to procure and assemble service sets complying with the solution to a given problem.

Returning to the example, the Obtain Book Service abstracts away any implementation choices or provisioning mechanisms. Hence, it brings the Library's production to an essential level that puts them all in the same level, which is the first step in allowing comparison to other alternatives of bringing about such item. Some examples are online ordering, loaning at a library or borrowing from a friend. Each of these variations introduces an offer at the solution market level with specific pricing and dependencies, which end up providing different end-user experiences.

While organizational-centric modelling may seem more natural because of its formal boundaries, the service structure is arguably more important since it is, by definition, focused in performance and value creation. This happens even at an intra-organizational level, as each sub-system is a means for providing value through services. Due to space constraints we will not define the structure of individual services in this paper, but we refer the reader to [11], where a framework is presented for service specification based on enterprise ontology.

3.2.3 Problem Solving Layer

Essentially, this layer is responsible for defining the problem statement and matching it to solutions available in the market. These solutions are sets of services that are contextualized and presented as value exchange propositions. In order to perform meaningful modelling and reasoning it is essential to establish the *purpose* as it is the

base for designing and engineering the solution providing system. Purpose is: '(...) an object or end to be attained; what one intends to do or bring about', according to the Merriam-Webster dictionary. A system's purpose is hard to formalize as stakeholders frequently formulate a high-level solution instead of the real problem, or present it in ways that induce specific solutions, such as in the classic example by Henry Ford: "*If I had asked people what they wanted, they would have said faster horses*".

Language and problem formulation is also critical as it drives the definition of the elements of the solution set [12]. An interesting model for its formal explicitation is presented in [13]. It consists of Need, Want and Demand structured in an hierarchy consisting in a transition from a *need* - a problem statement – to a high-level solution, defined as set of services that together provide a solution for that need - a *want* - and then to the formulation of a *want* in terms of value exchange proposition - a *demand*.

There are two other significant obstacles to problem solving, from the set identified by Mayer [14], that we are interested in tackling in this research:

Functional Fixedness: the tendency to view problems only in their customary manner, preventing vision over different options that might be available to find a solution. This is directly related to the upstream ramifications in a value chain.

Assumptions: when dealing with a problem, assumptions about the constraints and obstacles are often made, preventing certain solutions.

Both are conceptually addressed by using Reverse Discovery and recursion in the application of the problem solving to each engineering step.

4. Conclusion

This paper presents the current research results as an overview of a complex and largely subjective problem space. The presented structure and methodology is deliberately generally applicable to any human-engineered system, not only organizations, making it a very ambitious modeling effort in terms of abstraction.

During literature review, we were unable to find any framework structured in a way that solves the identified problems. We are confident that they are extremely relevant since they can be reiterated at any system/sub-system relation, either at pure business level, business/ICT interface or inside complex ICT systems. The abstraction and flexibility enabled by the recursive application are especially relevant in ICT-intensive environments, as the access to components usable as pieces of a solution chain is increased and maturing technological advances, such as the Cloud, make real-time service market start to look plausible in a relatively short timeframe.

Our contribution is composed by: 1) the identification of a relevant problem space in current approaches (both in academia and industry), particularly the lack of a sound structure to model purpose and serve as an ongoing referential, instead of addressing it solely at the early stages of individual system development cycles and losing track of it afterwards; and 2) the definition of a *conceptual high-level framework* that addresses, by design, the main issues identified in section 2 of this paper. It integrates the core concepts and their relative positioning in a layered manner, differentiating the concepts that characterize a problem/solution pair end-to-end, from need to implementation. The most important conceptual contributions are:

1 *Integrating the Teleological and Ontological perspectives* of system development by framing it in a *problem-solving context* and introducing the concept of *Market*;
2 Defining the *rationale of choices in terms of availability of solutions in a market*. This is accomplished by recursively defining purpose of a system as its contribution to that specific chain;
3 The *Reverse Discovery* concept as a different view over the GSDP, allowing structural accommodation of innovation dynamics.

Combining with Design Science Research, the methodology applied includes Action Research and has been adapted to a professional context in IS Demand Management, interfacing Business and IT at a leading Telco operator. Activities include analyzing motivation, impacts, cost vs. benefit, consolidation and planning of initiatives. Additionally, we have modelled part of the framework in formal ontology and build a Protégé-based prototype for supporting a preliminary case study, which has been used for instantiation of real world scenarios and was instrumental in eliciting hidden value assumptions obscured by upfront, unguided, service design.

References

1. Dietz, J.L.G., *Enterprise Ontology: Theory and Methodology*. 2006: Springer.
2. Bell, M., *Service-Oriented Modeling: Service Analysis, Design and Architecture*. 2008, New Jersey: John Wiley & Sons.
3. Lankhorst, M., *ArchiMate Language Primer*. 2004.
4. Lamsweerde, A.V. *Goal-Oriented Requirements Engineering: A Guided Tour*. in *International Symposium on Requirements Engineering*. 2001. Toronto.
5. Regev, G. and A. Wegmann, *Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering*, in *13th IEEE International Requirements Engineering Conference (RE'05)*. 2005, IEEE: Paris, France.
6. Gordijn, J., E. Yu, and B.v.d. Raadt, *e-Service Design Using i* and e3value Modeling.* IEEE Software, 2006. **23**: p. 26-33.
7. Gordijn, J., *Value-based requirements Engineering: Exploring innovatie e-commerce ideas*. 2002, Vrije Universiteit Amsterdam: Amsterdam.
8. Osterwalder, A. and Y. Pigneur, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*. 2009: Self-Published.
9. Aveiro, D., *G.O.D. (Generation, Operationalization & Discontinuation) and Control (sub)organizations: a DEMO-based approach for continuous real-time management of organizational change caused by exceptions*. 2010, UTL: Lisboa.
10. Beck, K., et al. *Manifesto for Agile Software Development*. 2001 [cited 2010-11-28]; Available from: http://agilemanifesto.org/.
11. Albani, A., et al. *Enterprise Ontology Based Service Definition*. in *4th International Workshop on Value Modeling and Business Ontologies (VMBO)*. 2009.
12. Schooler, J.W., S. Ohlsson, and K. Brooks, *Thoughts beyond words: When language overshadows insight.* Journal of Experimental Psychology, 1993(122): p. 166-183.
13. Kinderen, S., J. Gordijn, and H. Akkermans. *Reasoning about customer needs in multi-supplier ICT service bundles using decision models*. in *11th International Conference on Enterprise Information Systems*. 2009. Milan.
14. Mayer, R.E., *Thinking, problem solving, cognition*. 2nd ed. 1992, New York: W. H. Freeman and Company.