Value-Oriented Solution Development Process: Uncovering the Rationale behind Organization Components

João Pombinho[1,2], David Aveiro[3], and José Tribolet[1,2]

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

Abstract. Although significant progresses have been made in recent years regarding the goals of Enterprise Engineering, we find that the rationale behind every component of an organization is still not systematically and clearly specified. Indeed, state of the art approaches to enterprise development processes do not explicitly incorporate an essential dimension of analysis: value. This state of affairs does not warrant a leading role in enterprise alignment.

We propose to address this issue by specifying a value-aware system development process and a system development organization. To this end, we began by applying DEMO to model the system development organization. Furthermore, the original Generic System Development Process (GSDP) was modelled, and improvement points identified. Our main contribution is a revision of the GSDP, combined with research on value modelling and enterprise architecture that explicitly includes the teleological part of the system development process.

The explicitation of the development process focusing on the value dimension, contributes to providing traceability and clarifying the rationale of each organizational artefact. We believe that modelling this rationale systematically will improve reactive and proactive change management through increased self-awareness, improved scenario specification, objective evaluation and well-grounded system development decisions.

Keywords: DEMO, GSDP, Value Modelling, e3Value, Solution Development.

1 Introduction

Business complexity and environmental change pace coupled with increasing ICT support exponentially increases the entropy of business systems. The mechanisms humans use to manage the complexity inherent to these systems and their dynamics pose various challenges, as they are not based on transversal, coherent and concise models. At the same time, cost reduction through effective reuse, reengineering and

H.A. Proper, D. Aveiro, and K. Gaaloul (Eds.): EEWC 2013, LNBIP 146, pp. 1–16, 2013.

innovation being heavily demanded features from enterprises and their supporting systems. Laudon notes that enterprise performance *is optimized when both technology and the organization mutually adjust to one another until a satisfactory fit is obtained* [1]. However, studies indicate as much as 90 percent of organizations fail to succeed in applying their strategies [2].

Misalignment between the *business* and its *support systems* is frequently appointed as a reason of these failures [1, 3]. Aligning Business and IT is a widely known challenge in enterprises as the developer of a system is mostly concerned with its function and construction, while its sponsor is concerned about its purpose, i.e., the system's contribution. Also, the business vision of a system and its implementation by supporting systems is not modelled in a way that adequately supports the development and evolution of a system and its positioning in a value network. A paradigm shift in the way of modelling and developing systems must occur so that they can be increasingly developed considering their dynamic context and formally addressing the rationale behind value network establishment and system/subsystem bonding.

Formally integrating the notion of purpose into system development activities requires addressing both the teleological and ontological perspectives in an integrated, bidirectional way [4]. However, Engineering approaches are generally focused solely on the ontological perspective [5]. By Enterprise Engineering is meant the whole body of knowledge regarding the development, implementation, and operation of enterprises [6]. DEMO has a particularly relevant role in this area both as ontology and as a method. The Generic System Development Process (GSDP) is specified in DEMO's TAO-theory as the process by which a system is designed and implemented from the specifications of its using systems. The GSDP is systematically defined, clarifying normally ambiguous concepts like architecture, design, engineering and implementation. However, it lacks in instantiation and practical demonstration of usefulness.

This paper addresses the mentioned challenges by combining enterprise engineering and value modelling and is structured as follows: section 2 presents related work and the problem at hand. Section 3 introduces a practical scenario that will be used for reference through the paper. In section 4, we present our solution proposal and a more detailed instantiation of the method, with localized analysis. The paper closes with contribution summary and conclusions.

2 Related Work and Problem Statement

2.1 Related Work

In this section we introduce the enterprise engineering (EE) discipline and enterprise ontology and DEMO, a theory and method of EE. Next, we present e3Value, an approach to value modelling.

2.1.1 Enterprise Ontology and DEMO

Enterprise ontology [6] includes a sound theory and a method for supporting enterprise engineering. It goes beyond traditional function (black-box) perspective aiming at changing organizations based on the construction (white-box) perspective.

Organizations are considered as systems composed of social actors and their interactions in terms of social commitments regarding the production of business facts.

From the Transaction Axiom of Enterprise Ontology, we find that actors perform two kinds of acts. By performing production acts (P-acts), the actors contribute to bringing about and delivering services to the environment. By performing coordination acts (C-acts), actors enter into and comply with commitments. An actor role is defined as a particular, atomic 'amount' of authority, viz. the authority needed to perform precisely one kind of production act. P-acts and C-acts occur in generic recurrent patterns, called transactions. Every transaction process is some path through this complete pattern, and every business process in every organization is a connected collection of such transaction processes [6].

From the Distinction Axiom of Enterprise Ontology's PSI-theory, we find that we can divide all acts of an organization in 3 categories - ontological, infological and datalogical, respectively related with the 3 human abilities: performa (deciding, judging, etc.), informa (deducing, reasoning, computing, etc.) and forma (storing, transmitting, etc.). By applying both axioms, Enterprise Ontology's Design and Engineering Methodology for Organizations (DEMO) is able to produce concise, coherent and complete models with a dramatic reduction of complexity.

Unlike other approaches, DEMO makes a very strict distinction between teleology, concerning system function and behaviour – the black-box perspective – and ontology, about its construction and operation – the white-box perspective [7]. These perspectives are embodied in the Generic System Development Process (GSDP), represented in Figure 1. It begins with the need by a system, the Using System (US), of a supporting system, called the Object System (OS).

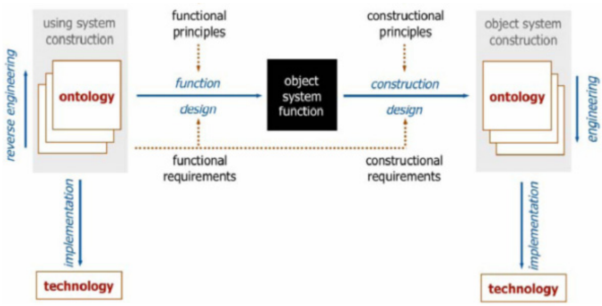

Fig. 1. Generic System Development Process [6]

From the white-box 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 white-box 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. However, nothing is prescribed about the rationale behind these choices. System design decisions, either implicit or explicit, remain solely, and certainly not forever, in the minds of the participants in the process. The sheer complexity can quickly cross the limits of unsupported human handling. It may then become short of impossible to know the rationale of past decisions, its impacts and dependencies in designing the to-be.

2.1.2 Developing Organizations with Control and G.O.D Sub-organizations

Aveiro took a step towards instantiating the GSDP by applying DEMO to specify the models of the sub-organizations responsible for handling change caused by exceptions. In the control sub-organization [8], the viability of a system is specified by a set of measures and respective viability norms that can be periodically checked against the operational status. If such norms are violated, a dysfunction handling mechanism is triggered. If the exception that causes the dysfunction to the norm is expected, solutions that have previously been identified in anticipation are applied and evaluated for solving the problem. If the cause is unexpected, an organizational engineering process (OEP) must be started, that occurs in the scope of another sub-organization, the G.O.D. organization [9], responsible for specifying and implementing change that will solve or circumvent the unexpected exception causing a dysfunction. The solution may be new organizational components (e.g., new norms, new actors, processes and rules, etc.) or just (re-)allocation of human or IT resources.

2.1.3 Value Modelling – e3Value

There are many classifications of organizations, according to their composition and objectives, including: private, public, political, business, educational, healthcare, non-profit, etc. All organizations have in common bringing about *value* to their *environment*, either directly or indirectly, so *value* is an unifying concept. Also, Value Modelling was selected as it is increasingly recognized that the concept of value assists in improving stakeholder communication, particularly Business and IT [10].

e3Value [11] is part of e3family, a set of ontological approaches for modelling networked value constellations. It is directed towards e-commerce and analyses the creation, exchange and consumption of economically valuable objects in a multi-actor network. In e3Value, an Actor is perceived by his or her environment as an economically independent entity, exchanging Value Objects. An enterprise is modelled as an actor in a value network, where the demand and offer market concepts are a natural consequence of the economic context of Value Objects.

As will be presented in section 4, we propose applying e3Value to improve system and subsystem value modelling: inside the boundaries of organizations, as opposed to applying it solely to e-commerce relations between formal organizations.

2.2 Problem Statement

Looking at previous efforts on formalizing organization development, one question that comes to mind is: what are the criteria for generating new organizational components? In [12] generic acts of monitoring, diagnosis and recovery are used to specify the rationale behind change. But such categorization is quite generic and does not explicitly capture an essential dimension of analysis: value. As an example, we can think of a viability norm as the minimal number of movie loans per month at a video store. In practice, this is an economic condition for having minimal profit required for sustained survival and growth of Watch-it business, the generic and main value condition for the company. However, if only a "local" perspective is taken during viability norms specification, global, combined effects of these and other value drivers are missed. Still, broader rules can be applied and the combined effect of drivers can be calculated and set as a wider rule. But even so, the very structure of the organization and the reasoning behind these rules may not be precisely captured.

We hypothesize that these rules are set during the implementation of not only reactive (the focus of GOD) but also proactive and evolutive changes of the organization. Such rules must not only conform to but justify its structure as there is a bidirectional relationship between value conditions, value network and the organizational structure as well as the resources needed to "implement and run it".

During a system development process, the designed system/subsystem relations are a result of choices between different solutions for intermediate and possibly interconnected sub-problems. Such sub-solutions can and should be modelled as individual system development efforts, preserving the modularity that allows for rigorous modelling and tracing of the rationale behind these intermediate choices. By defining a formal model of the development process, the relations between systems and sub-systems can be made explicit as problem/solution pairs, thereby explicitating the nature of these relations and flattening the system structure, while preserving rational structure as it will be explained in section 4.2.

In order to clarify our solution proposal to these issues, we chose to model the system development organization. It seemed appropriate to apply DEMO to the GSDP itself, as a system development organization, and defining its own ontological model. The results were then combined with previous research on value modelling [4, 13].

In the following sections we explore the reasoning just presented and research results in two phases. The first, intended as a formalization of the GSDP as defined in [6]. The second phase is a revision of the GSDP to include the teleological part of a given system development process.

3 Unimedia Case: Remote Internet Customer Support

Unimedia is a quadruple-play operator (television, internet, fixed voice and mobile voice) with a large customer base. Customers may have a combination of services and some services require customer premises equipments (CPE). These equipments amount to a relevant part of customer support, particularly for the internet service. The remote customer support organization is described by the following narrative:

In the case there is a perceived malfunction by the customer, she can contact the call center directly to identify and solve the issue. After calling the support number, her call is handled by an Interactive Voice Response (IVR) system. IVR allows customers to interact with the company via telephone keypad or by speech, so they can service their own inquiries by following the predefined process or, eventually, get redirected to a human operator. The client identifies by dialling the national ID number. Additional identification information can be requested for cross-check later in the call if *there are relevant actions to be taken. Following, a diagnosis process is carried on. The diagnosis can be at customer side (e.g. check the modem lights) or at the provider side (e.g. check service provisioning status). After a diagnosis is established, a solution is attempted. Again, the solution can be at the customer side (e.g. reset device) or at the provider side (e.g. force firmware update). The call ends after reaching a solution or, if it is not successful, by requesting field service.*

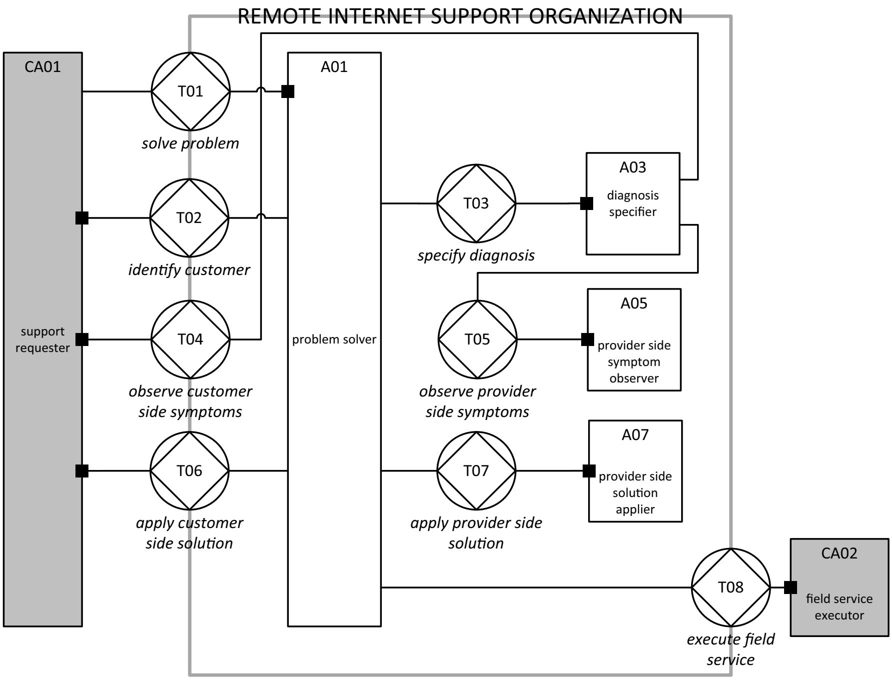

Fig. 2. Unimedia Remote Internet Support Actor Transaction Diagram (ATD)

Following the alignment process described in [14], an extension to the Transaction Result Table (TRT) was proposed, including the concepts of Value Object (VO) and Value Transaction. The resulting value model is shown in Figure 3.

Table 1. TRT extended with Value Object and Value Transaction

Transaction kind	Transaction result	Value Object	Value Transaction
T01 solve problem	**R01** problem P has been solved	P solution	**VT01** Problem solution for Money (contract)
T02 identify customer	**R02** customer contract C has been identified	C contract	**VT01** Problem solution for Money (contract)
T03 specify diagnosis	**R03** diagnosis D has been specified	diagnosis	**VT02** Diagnosis for Money
T04 observe customer side symptoms	**R04** customer side symptom CSS has been observed	CS symptom	**VT03** CSS for Eligibility
T05 observe provider side symptom	**R05** provider side symptom PSS has been observed	PS symptom	**VT04** PSS for Money
T06 apply customer side solution	**R06** customer side solution CSP has been applied	CS action	**VT05** CSP for Eligibility
T07 apply provider side solution	**R07** provider side solution PSP has been applied	PS action	**VT06** PSP for Money
T08 execute field service	**R08** field service FS has been executed	FS solution	**VT07** FS for Money

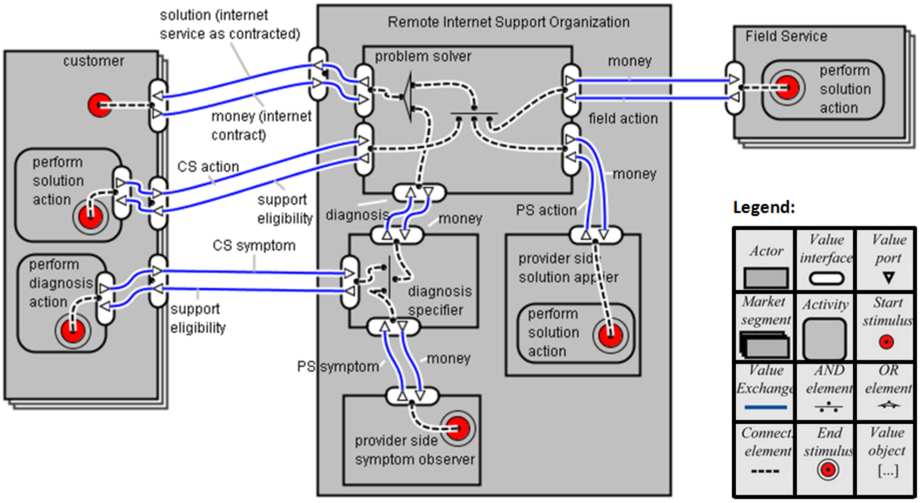

Fig. 3. Value model for Remote Internet Support scenario

The description of the process and benefits of aligning value and ontological models exceeds the scope of this paper and is presented in [13, 14]. Still, a brief example of contributions from both sides follows.

Some clarifications resulting from aligning ontology models based on value were the explicitation of value activities. For instance, as part of getting free remote customer support, the customer must provide "eyes & hands" to get *support eligibility*, which is the VO. Actually this is company policy but was missing from the narrative and was identified due to the notion of *economic reciprocity* from e3value – the transactions must have at least an inbound value port and an outbound value port. Also, note that *CS symptom* and *CS action* are relevant VOs because they are intermediate results for their respective solution chains: diagnose problem and solve problem.

On the opposite direction, the main contribution of ontological analysis is that social interaction theory and, particularly, the transactional pattern allow checking the value model for completeness and consistency. One example is testing the value object exchange over the complete transactional pattern, with possible impacts on (re-) specification of value objects and interfaces, e.g., what happens if a customer declines performance of local diagnosis?

4 Improving the GSDP - Introducing Purpose and Value

4.1 Applying DEMO Methodology to the GSDP

We define a solution to a problem as the production of a determined result, which generally involves investment of resources (time, money, effort, etc.) by the Object System (OS) and generates value for some stakeholder, the Using System (US). By asking the solution requester to define the construction of the US and its value model, additional insight can be derived from its specification. This insight can change the problem or dissolve it altogether. However, the entry point of the GSDP, i.e., the origin of the system development request, is not sufficiently clear in the original model. To overcome this issue, we defined the Solution Development Organization (SDO), presented in Figure 4.

In our view, the description originally provided for the GSDP was not ontologically complete and some adjustments were in order to obtain a coherent model of the SDO. Particularly, we defined a recurrent *provide solution* transaction (N+1) as a new solution development cycle where the current OS assumes the role of US and a new OS is being developed so that its function serves the construction of the US. This transaction is represented by the link between A03 and T01 and is crucial for explicit multi-cycle solution development, i.e., function/construction alternation.

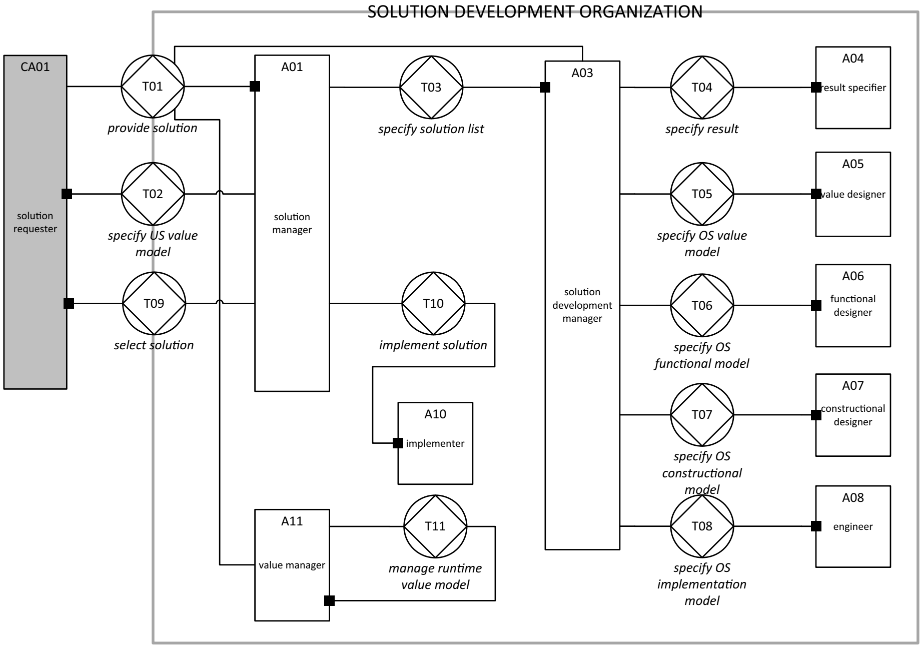

Fig. 4. Solution Development Organization – ATD

The process begins with an external request to provide a solution. In this case, Unimedia's Head of Customer Support requests a solution for reducing costs, following a decision by the board that their internet support costs are to be reduced by at least 20%. The solution manager asks the requester to specify the Using System value model, which is critical to identify rational solutions. In this case, the requester produces a value model, showing that the largest costs come from the calls that get redirected to human operators. The solution manager then requests that the solution development manager specifies a solution list to produce the result requested, in the context of the US value model. The specify results transaction is the creative step of this process, where different ways of producing the required result (solution) are identified. For instance, one idea would be to recruit cheaper operators; another would be that the less calls were redirected to the human operators. For each result, the value model and functional model are specified in sequence. Next, the constructional model is built, where transactions and actors are specified. In this case, the result would be to lower the number of redirections to expensive human operators by 20%.

If there is a dependency in producing the result, then another solution development process is triggered, with the solution development manager requesting a solution for that problem. The current OS is repositioned, assuming the role of US in the new development cycle. For instance, the dependency can be to find a solution to provide additional checks and redirections to avoid costly human operators whenever possible. Such a request would be made by the level 1 solution provider to level 2 providers. For each crossing of these levels, a new GSDP iteration takes place. Along each single thread of a solution chain, the alternation between each pair of levels is described by Dietz and Hoogervorst as function/construction alternation [6]. A set of such iterations is commonly performed implicitly inside a single GSDP, and thereby kept from being adequately modelled by the explicit application of functional, constructional and architectural principles.

When the set of known solutions is considered satisfying by the solution manager, it requires that the solution requester elects a solution from the presented alternatives. The elected solution is implemented and its value proposal is periodically monitored by the value manager. If an inconsistency is found, the *provide solution* transaction is invoked to address the gap, presented as an economic viability problem.

4.2 The Method at Work: Value-Driven Cost Reduction

We now present the method inherent to the solution development organization. This generic method applies to both a bootstrapping setting or to an ongoing change.

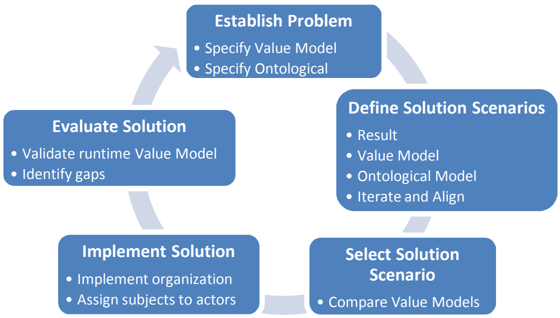

Fig. 5. VoSDP - Method for practical application

I - Establish Problem

Revisiting our example, let us begin with the initial request. The fact that the investors are the requester means they must come into play explicitly in the value model. The first step is to represent the *as-is* set. Due to space limitations, a simplified generic value model of a private, for-profit enterprise is presented in Figure 6.

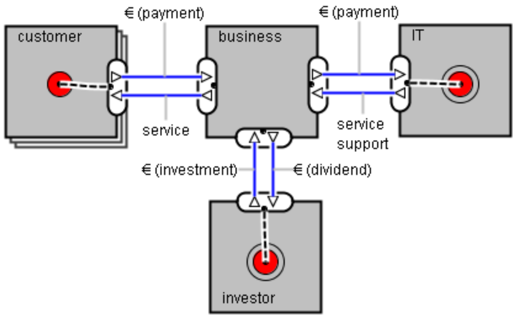

Fig. 6. Generic value model of IT-enabled for-profit enterprise

Considering the value model in Figure 6, the request can be reformulated as improving the investor balance (the real sought-after result). In this simple model, the equation for an annual period is:

$$dividend = customer\ revenue - business\ OPEX - IT\ OPEX - investment$$

The result can be attained by reducing the expenditure or finding alternative ways of generating value, such as increasing revenue (relating with the customer actor) or decreasing support costs (relating with the IT actor). Somewhat surprisingly, as we are about to see, the choice was increasing investment and IT OPEX costs.

II - Define Solution Scenarios

After clarifying the problem, the solution manager starts a solution development cycle that returns a list of possible solutions in a reasonable period of time. Please note that modelling the value of the solution development process itself and, therefore, obtaining a consolidated value model that takes into account return on modelling effort (ROME) can be done by using the same methodology but exceeds this paper's scope.

One obvious solution, which is exists in most situations, is to leave everything as it is. By default, this represents the baseline scenario. The solution development manager is to identify additional scenarios and begins as a mostly creative endeavour of identifying results/value objects that make up the following nodes on the value chain for obtaining the original result. In our example, it is necessary to know the cost structure of the *business* actor from Figure 6. For simplicity and space economy, let us consider that Figure 2 is a complete model of the business actor. In this case, it turns out that the *problem solver* stands out by a large margin while analyzing the transactional costs of the actors (part of the e3Value model). The fact that the ontological model of the organization does not allow concluding this is no surprise as it abstracts implementation. On the other hand, the value model also has an invariant perspective but it is complemented with selected implementation-level constructs. In this perspective, it is possible to include parameters that are implementation-level value estimates.

Returning to the example, the transactional costs of the problem solver are mostly due to the time the human agent spends in 1) initial call handling, i.e., identifying the customer and 2) filtering away exceptions to normal diagnosis.

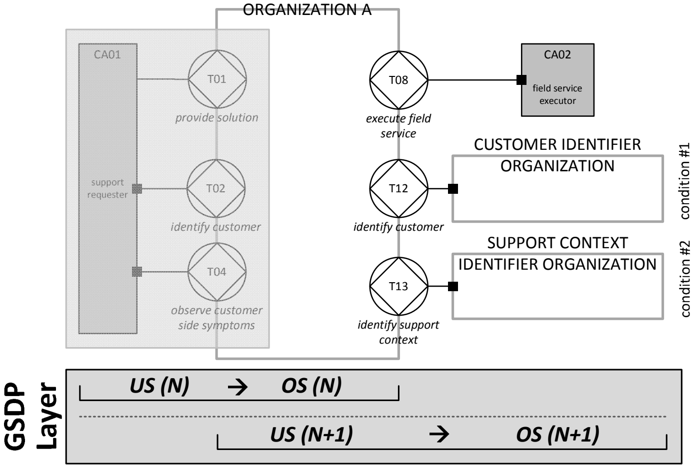

Fig. 7. Recursive SDP cycles for addressing identified problems

As represented in Figure 7, two (layer 2) system development processes (SDP) are started by the *solution development manager* (layer 1) with the request of reducing human operator time by each of the conditions mentioned above. The value model is already clarified, as it previously existed at an adequate detail level to specify the problem scope in a way that can be related to the ontological model.

The problem related to condition 1 can be solved by using existing CRM services to identify a customer by a set of keyed-in data that is sent to the IVR system via Dual-Tone Multi-Frequency (DMTF).

The problem related to condition 2 can be solved by using existing services that, based on the customer address and portfolio, can identify situations that constitute exceptions to the scope of the remote internet support system. For instance, Service Management ticketing system services can identify if there is a common problem in the geographical area of the customer service address/cell, e.g., a power blackout or a cut backbone. In these cases, there is no added value in handling the call to a human operator, so an automated vocal message that describes the situation and provides expected resolution time allows ending the call without resorting to a human agent.

Note that, in the case of condition 1, the *identify customer* transaction already exists and, despite being executed by the customer it is still problematic. The issue is in the complexity of the accept phase of the transaction pattern. By splitting the initiating actor of the transaction, it is now possible to allocate different subjects that can execute more efficiently. It can be argued that these solutions are simple automations and do not change the construction, being solely implementation choices. We are aware that these conditions are of infological nature but to implement each solution, it is necessary to add an actor to the construction. This actor has its own business with service levels and responsibilities, which is the same as saying we are dealing with a US and OS both at the B-level so it is a matter of relativity, as discussed in [4].

III - Select and Implement

In order to rationally select solutions scenarios, objective criteria must be defined. To this end, using e3Value it is possible to assign valuation formulas to value object transfers through value ports. There are two types of value objects: 1) money objects, when the amount transferred can be objectively stated and observed; and 2) non-money objects: the value is subjective, meaning actors can disagree about the amount of economic units they assign it.

While non-money objects can be important for design and impacts analysis, there should be an effort to monetize costs/benefits to allow financial analysis. The techniques and theory for doing so are out of the scope of this paper. Nevertheless, it is worth noting that, seemingly, the value model creation by itself is a step forward creating consensus and improving objectivity.

Besides the valuation of individual transactions on each value port, e3Value defines the concept of expenses which contribute to the economic viability analysis but are not explicitly modelled as value exchanges (e.g., employee costs).

- Variable expenses – occur multiple times per value model, depending on the transaction volume and are assigned to value ports, being useful for modelling operational expenditure (OPEX);
- Fixed expenses – occur only once per period, e.g. monthly wages, used for simplification, also useful for modelling OPEX;
- Investments – a particular fixed expense, occurring only once per time series (scenario) and therefore useful for modelling capital expenditure (CAPEX).

e3Value allows specifying value model components using specific attributes that make the profitability sheets directly derivable from the model. Table 2 and Table 3 represent simplified annual profitability sheets for both scenarios, where value port details have been excluded.

Table 2. Profitability sheet for scenario A

Actor	Interface	Occurences	Valuation	Value
problem solver	problem solution, internet contract	1080000	0,30 €	- 324.000 €
	EXPENSES			- 5.000 €
	INVESTMENT			- €
actor total				- 329.000 €
overall				- 329.000 €

Table 3. Profitability sheet for scenario B

Actor	Interface	Occurences	Valuation	Value
problem solver	problem solution, internet contract	918000	0,24 €	- 220.320 €
	EXPENSES			- 5.000 €
	INVESTMENT			- €
actor total				- 225.320 €
customer identifier	problem solution, internet contract	1080000	0,02 €	- 21.600 €
	EXPENSES			- 1.200 €
	INVESTMENT			- 15.000 €
actor total				- 37.800 €
support context identifier	problem solution, internet contract	1080000	0,01 €	- 10.800 €
	EXPENSES			- 1.200 €
	INVESTMENT			- 15.000 €
actor total				- 27.000 €
overall				- 290.120 €

For the sake of illustration of the concepts of this paper, we chose to represent only the cost stream, assuming stable customer revenue. Out of the 1,5M customers, based on historical data we considered that 900K have internet services and they call for support 1,2 times/year on average. We assume the implementation of the new actors has a CAPEX of 30 K€ and OPEX of 2,4 K€, allocated in halves to each. After implementation, the number of calls handled by problem solver has reduced by 15% and the average support call duration for this actor lowered from 10 to 8 minutes because of effectively reduced support scope due to early context clarification.

In this particular situation, the option is clear after comparison of the value models. Selecting scenario B is relatively straightforward, as there is an interesting business case versus scenario A. Scenario B represents an investment with a payback of 6 months and, onwards, a positive impact of 5,7 K€/month versus scenario A. It is noteworthy that the existence and relevance of an *investor* actor is formally required because it is included in overall value model. Obviously, there may be different scenarios and additional analysis with impact on their definition may be called for, resulting in additional iterations. For instance, the investor may try to find better solutions to invest his money and get payback in less than 6 months.

In this example we left out using time series, a concept that directly addresses the time variable and establishes value models for specific consecutive time periods. This view is useful not only for business case specification over time but also to align expected value production with solution architecture and construction roadmaps.

The relevant aspect of implementation we want to make clear, besides its technicalities, is that the implementation of the artefacts is also accompanied by putting the business model itself into operation (production environment). We refer to this as a *live business model*, in opposition of using it solely for evaluation and decision purposes early in the process. This means that the value model is now an artefact which is controlled by a specific actor, *value manager*. The *value manager* compares operational reality to the specification in the model and may decide to request the problem solving organization to address a potential gap. While the detailed specification of how this comparison is carried through is out of the scope of this article, it is relevant to note that it is enabled by the existence of specific constructs to model value.

IV - Evaluation

Evaluation happens both at the implementation review of a project and continuously at runtime, in the spirit of the *live business model* concept. The exhaustive description and analysis of this phase exceeds the scope of this paper, but it can be concluded that the explicitation of the development process and the intermediate deliverables produced contribute to the availability and objectivity of evaluation mechanisms. In our example, if the implementation of the services that provide information for identifying support context is consistently unreliable, the projected benefits may not be achieved and may even have negative results because of mistaken call redirections. The advantage of having a business case integrated with the ontological and implementation models is that it is now possible to estimate the critical values that put economic viability at stake and monitor them in anticipation via trend analysis.

Evaluation can also lead to exploring alternative ways that were not selected but that were considered at an earlier phase of the solution development process. Leveraging the prospective solutions concept presented earlier lets us, e.g., return to the original solution request and the idea of increasing revenue. One way of contributing to this value stream is by reusing the automated IVR-based solution just developed as a channel for up selling/cross-selling. This repositions the customer care organization from a cost center to a value center. The opportunity of having the customer in-line can be taken advantage of by creating a discounted offer for these situations, to be presented automatically (relatively inexpensive) and/or redirect the call to a sales operator (more costly; more effective?). To explore this path, a new GSDP cycle is in sight. Only after successful solving the customer problem, of course!

5 Conclusion

We found that in order to capture the rationale behind organizational artefacts, we need additional constructs to those DEMO currently provides. The contributions of this paper can be summarized as redesigning the GSDP and the corresponding SDO for supporting multiple cycles and extending it with value concepts. Alternating Value/Function/Construction in successive cycles was found relevant and applicable.

As explained on section 3, the main contribution of ontological analysis to the match with value modelling is that social interaction theory and, particularly, the transactional pattern, allows checking the value model for completeness and consistency. Conversely, by integrating value modelling with ontological modelling we can anticipate decisions based on projected implementation viability and leave a formal trace of the decision rationale. Moreover, we exemplified how the resources used in the implementation of the system may relevantly restrict the ontology of the system: 1) there are ontological subsystems purely constructed by some value condition and 2) the value specification must be part of the production world. Very frequently, parts of the construction depicted in the ontological model depend on value constraints at implementation level and to strive for fully implementation independent models would be either unfeasible or a simplistic approach with unuseful models as a result. Therefore, we see ontological models not as implementation *independent*, but rather implementation *abstracted*.

The benefits from these contributions go beyond the simple support system automation and rest on the capacity to model the essentials of the businesses involved in their commonalities and differentiators. Each variation point of a business area places demands on the construction of the organization providing these services. These services are valued distinctly by different customer types and this value should be actively managed in articulation with the construction. In turn, they allow exploring synergies through reutilization of solutions and increased insight given by explicitating the intermediate artefacts of the solution development process.

For all scenarios considered, even if the solution development step is not completed for some reason, e.g., lack of investment capability or analysis time, every deliverable is kept in association with the problem specification. While this is somehow obvious for complex deliverables, such as value models, even a simple enumeration of results in a hypothetical chain, with generalization or specialization of the value objects, represents prospective solutions that can be revisited later on.

As it can be seen from the example, there is no magic bullet regarding creative solution hypothesizing. As a practical observation and clarification, our method allows domains experts to be involved by the responsible actors in both the solution development and selection transactions. Some mechanisms based on knowledge about prospective or used solutions, for instance generalization/specialization of value objects may be used as a starting point. Still, there is no greater ambition than to provide useful tools for the human mind to do its job.

References

1. Laudon, K.C., Laudon, J.P.: Management Information Systems: Managing the Digital Firm. Prentice Hall (2011)
2. Kaplan, R.S., Norton, D.P.: Strategy Maps: Converting Intangible Assets Into Tangible Outcomes. Harvard Business School Press, Boston (2004)
3. Henderson, J.C., Venkatraman, N.: Strategic alignment: leveraging information technology for transforming organizations 32(1), 4–16 (1993)
4. Pombinho, J., Aveiro, D., Tribolet, J.: Towards Objective Business Modeling in Enterprise Engineering – Defining Function, Value and Purpose. In: Albani, A., Aveiro, D., Barjis, J. (eds.) EEWC 2012. LNBIP, vol. 110, pp. 93–107. Springer, Heidelberg (2012)
5. Op 't Land, M., Pombinho, J.: Strengthening the Foundations Underlying the Enterprise Engineering Manifesto. In: Albani, A., Aveiro, D., Barjis, J. (eds.) EEWC 2012. LNBIP, vol. 110, pp. 1–14. Springer, Heidelberg (2012)
6. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer (2006)
7. Dietz, J.L.G.: Architecture - Building strategy into design. Netherlands Architecture Forum, Academic Service - SDU, The Hague, The Netherlands (2008)
8. Aveiro, D., Silva, A.R., Tribolet, J.: Control Organization: A DEMO Based Specification and Extension. In: Albani, A., Dietz, J.L.G., Verelst, J. (eds.) EEWC 2011. LNBIP, vol. 79, pp. 16–30. Springer, Heidelberg (2011)

9. Aveiro, D., Silva, A.R., Tribolet, J.: Extending the Design and Engineering Methodology for Organizations with the Generation Operationalization and Discontinuation Organization. In: Winter, R., Zhao, J.L., Aier, S. (eds.) DESRIST 2010. LNCS, vol. 6105, pp. 226–241. Springer, Heidelberg (2010)
10. Cameron, B., Leaver, S., Worthington, B.: Value-Based Communication Boosts Business' Perception of IT. Forrester Research (2009)
11. Gordijn, J.: Value-based requirements Engineering: Exploring innovatie e-commerce ideas. Vrije Universiteit Amsterdam, Amsterdam (2002)
12. 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. UTL, Lisboa (2010)
13. Pombinho, J., Tribolet, J.: Modeling the Value of a System's Production – Matching DEMO and e3Value. In: 6th International Workshop on Value Modeling and Business Ontology, Vienna, Austria (2012)
14. Pombinho, J., Aveiro, D., Tribolet, J.: Business Service Definition in Enterprise Engineering - A Value-oriented Approach. In: 4th Workshop on Service Oriented Enterprise Architecture for Enterprise Engineering, Beijing, China (2012)