

# A Value-Oriented Approach to Business/IT Alignment – Towards Formalizing Purpose in System Engineering

João Pombinho<sup>1,2</sup>, David Aveiro<sup>3</sup>, and José Tribolet<sup>1,2</sup>

<sup>1</sup> INESC-ID and CODE - Center for Organizational Design & Engineering, INOV,  
Rua Alves Redol 9, Lisbon, Portugal

<sup>2</sup> Department of Information Systems and Computer Science, Instituto Superior Técnico  
Technical University of Lisbon, Portugal

<sup>3</sup> Exact Sciences and Engineering Centre, University of Madeira, Funchal, Madeira, Portugal  
jpmpe@ist.utl.pt, daveiro@uma.pt, jose.tribolet@inesc.pt

**Abstract.** It is widely recognized that a large percentage of IT initiatives fail from a business perspective. This is attributed to many factors, namely system complexity and change pace. We believe that the system development process itself is a crucial aspect of this state of affairs and a paradigm shift is required. There is a lack a common set of concepts and language to use through an IT development process. Essentially, appropriate models and founded theory for articulating the teleological and ontological perspectives of a system are necessary. In this paper, we present and discuss an innovative value-oriented approach to System Design and Engineering. Our contribution begins by identifying a relevant problem space regarding current approaches, particularly the lack of a sound structure to model a service system's *purpose*. We believe that system modeling with a market mindset will help improving quality and improve change response. The approach draws from a combination of theory based on Enterprise Engineering, Service Science and Value Modeling. A four-layer framework (System, Service, Value and Purpose) is pointed as a conceptual solution for simultaneously representing relevant concerns for promoting dynamic alignment between Business and IT.

**Keywords:** System Design and Engineering; Enterprise Engineering; Value-orientation; Purpose.

## 1 Introduction and Motivation

The current global economic crisis context is both an opportunity and an enormous challenge for businesses. Cost reduction through effective reuse, reengineering and innovation being heavily demanded characteristics 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]. Misalignments 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. We believe it is mainly due to an essential misalignment: 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. 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].

However, even if this problem was solved, an additional challenge mounts as enterprises are complex systems generally operating in competitive environment. Both their structure of enterprises and the processes that deal with their lifecycle need to balance investments in readiness for change and the effective usage of those investments in terms of value and frequency.

A strong assumption generally hinders solutions: formal organizations are generally created as providers of a *repeatable* and *stable* solution to a given demand. In this context, stable means that there is reasonable belief that the *elements* providing a solution will be continuously available – they are even considered part of the organization. The reason behind this quest for stability is, essentially, the lack of agility in procuring resources on-demand. This leads to compromise between evaluating different solutions to support each business iteration and the time and effort consumed in doing so. It may be argued that the main issue presented can be partially circumvented by using better implementation processes or by increasing modeling coverage and detail. However, we believe its origin is essentially structural. There should be a paradigm shift improving system development with change support in mind. Ideally, a framework should explicitly include the concept of market, with demand/offer dynamics, to address different and innovative solutions to business activity support.

This paper analyzes Business/IT alignment as *system/supporting system* alignment and is structured as follows. Section 2 presents research scoping, with Enterprise Engineering as ground for the Library example that illustrates problem analysis. Then, current challenges are identified and grouped into three problem areas, with a brief related work review. In Section 3 we present a set of core principles and an overview of our Framework. The paper closes with conclusions and contribution summary.

## 2 Problem Space

### 2.1 Scope: Systems, Enterprises and Value generation

*System* has its etymology in the Latin *systema*, *whole compounded of several parts or members*. Skyttner [6] defines it as ‘*set of elements arranged in such a way that they produce a recognizable outcome. It is also constituted by a Frontier with the external world, called its Environment*’. The formal definition we will use, from Enterprise Ontology [7], defines the following properties: *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 elements in the composition, and between them and elements in the environment.

This research addresses *engineered systems*, which are systems deliberately created or changed by means of engineering activities, i.e., with a purpose and rationale. This leaves out of scope naturally occurring geological systems, for instance.

Enterprise Engineering has as a premise that enterprises or, more broadly, organizations, are systems and therefore can be object of systems engineering activities. Organizations have many tipifications, according to their composition and objectives, including: private, public, political, business, educational, healthcare, non-profit, etc. All of these kinds of organizations bring about *value* to their *environment*, either directly or indirectly, so *value* is an unifying concept.

In this paper we present and discuss an innovative, value-oriented approach to System Design and Engineering. It combines work in related domains, such as Enterprise Engineering [7], Service Science [8], and Value Modeling [9].

## 2.2 Base Theory – Enterprise Engineering and Service Science

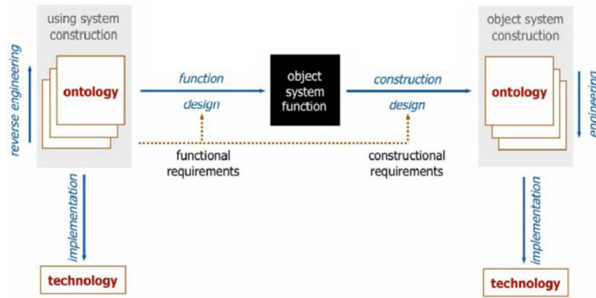
### Design and Engineering Methodology for Organizations (DEMO)

DEMO [7] is a cross-disciplinary theory for describing the structure and action of organizations, modeled as discrete dynamic systems consisting of social actors. These actors enter to and are responsible for coordinated commitments with each other. 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 actors and abstracts away implementation issues.

The *distinction* axiom concerns the separation of knowledge on a specific enterprise in three groups: *ontological* (*B*), *infological* (*I*) and *datalogical* (*D*). These are directly related with the abilities *performa* (deciding, judging, etc.), *informa* (deducing, reasoning, computing, etc.) and *forma* (storing, transmitting, etc.). This distinction is very important for distilling the essence of the organization as a social system, with a reported simplification of about 90% in model complexity [10].

In DEMO, an *organization* is defined as a social system, made up of *subjects*, who perform two kinds of *acts*: *production* (P-acts) and *coordination* (C-acts). An *actor* is a subject fulfilling an *actor role*. Actor roles abstract a particular subject performing an act, thus representing the *authority* to perform a particular P-act and related C-acts. *Transactions* are patterns of coordination acts performed in steps: *request*, *promise*, *state* and *accept*. In addition, the cancellation steps of these acts are represented: *decline*, *quit*, *reject* and *stop*. The transaction pattern has the particularity of being able to represent as a path every conceivable transaction. For this reason, it is a good model for the coordination of dual-party interactions between social actors.

To close this brief presentation of DEMO's base theory, we should distinguish two aspects of a system: *teleological*, concerning its function and behavior, a black-box; and *ontological*, about its construction and operation, a white-box [11]. This important distinction, generally absent from other state of the art approaches to our problem, forces both 1) the separation of these concerns and 2) their articulation. The *Generic System Development Process* (GSDP), shown in Fig. 1, addresses the hinge point between teleological and ontological modeling, beginning with the need by a system, the *using system* (US), of a supporting system, the *object system* (OS).



**Fig. 1.** Generic System Development Process. Adapted from [7]

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.

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.

### Service Science and Service-Dominant Logic

Both Service Science and Service-Dominant Logic [8] have been around for some years now, rising from the need of a fresh approach to how business is performed. The concept of *service system* [12] is central to both service science (SS) and service-dominant (SD) logic. It is defined as “a configuration of people, technologies, organization and shared information, able to create value to providers, users and other interested entities, through service”. Service as a process involves using an actor’s resources for the benefit of serving another actor. Social systems are, therefore, of paramount importance in SD-Logic and a conceptual compatibility point with system design and engineering. Additionally, the transaction pattern is common ground between DEMO’s transactional pattern and the Service Science concept of Service System, the Interact-Serve-Propose-Agree-Realize model [12]. Analysis of their mapping, presented in [13], allowed concluding that DEMO is quite comprehensive in modeling the flows between these interactions states.

Combining Service Science and Enterprise Engineering is proposed in [13] through the identification of convergence principles. Service Science provides economic theory concerning service exchanges between agents; Enterprise Engineering contributes with the knowledge on designing and engineering those systems.

### 2.3 Library Example, Issues Identification and Analysis

In order to clarify the problem space, a practical scenario based on the classical DEMO Library case [7] will be used for instantiation.

The Actor Transaction Diagram (ATD) of the Library is presented in Fig. 2. In this example, the elements of the system dealing with the library membership (solid line-bounded area in Fig. 2) are not justifiable as directly bringing 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:

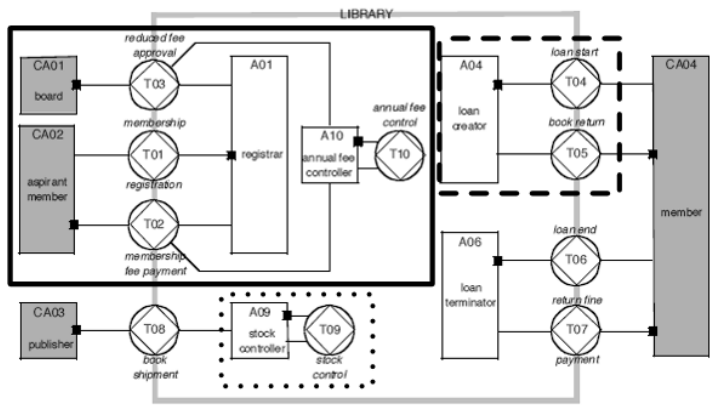


Fig. 2. Library ATD – Construction analysis

Having as a reference the Library’s core business, *providing reading content*, some remarks about modeling issues follow: 1) the core service is obscured inside a *loan transaction*, in the area marked by a dashed line; 2) the area bounded by points encloses a support process (stock control) that may need revision, for instance, in a scenario of going digital; 3) inside the solid black line, a *sacrifice* of the customer in obtaining the service and its support (sub)system, let us name it the *Membership Management* subsystem. Considering the later, one must ask if there is really a customer who *wants* a membership? Or was this subsystem included in the Library as a strategy to obtain 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 essential Library business concept, i.e., every library also offers it by definition? Under what conditions its value to the Library should be reviewed?

By analyzing these questions, three main problem areas were isolated:

**Defining Value.** Value is, by nature, dependent on the stakeholder and, thus, relative. The problems in adequately naming and scoping a service, known in the Service Design community, are symptomatic [14]. 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 interesting *per se*, or is it only in the way of getting a book, that is specific to this particular construction of a library? How does a Library compare to a Bookstore or a Publisher, from the customer’s perspective? This is why

current goal-oriented modeling [15, 16] 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, as explained on section 2.4.

**Supporting System (de)construction.** A system's *construction*, resulting from the development process, is a compiled structure that obscures system/subsystem relations and their motivation. Separating a subsystem from its owner system is hard, especially if it was modeled from a flat description of the operation of the organization instead of an incremental design step. Also, 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 solution scenarios, in order to provide flexibility points instead of a static solution path. Modeling these flexibility points requires the capability of decomposing a system into a chain of value-providing elements.

**Modeling system intervention rationale.** Implementing a system consists in introducing restrictions on its construction, for instance: 1) assumptions, such as assuming the customer is necessarily a reader; and 2) constraints, such as available technology to offer books, e.g., physical or digital. Still, the rationale of a change is not commonly kept in a reusable way. As an example, regarding the introduction of the Membership subsystem: 1) *What was its purpose? Was it for mitigating the risk of non-return?* 2) *What were the alternatives?* 3) *What were the design principles, constructional principles, assumptions and constraints applied? Do they still hold? Will they hold for any kind of content the library may want to provide, e.g., e-books?* DEMO has been extended [17] to incorporate change dynamics but, currently, still does not model the formal rationale of each change. The GSDP also does not prescribe what to do with the objects supporting intermediate decisions made during the process.

## 2.4 Related Work – State of the Art Approaches

**Enterprise Architecture – Archimate.** Archimate is an Enterprise Architecture modeling language with broad practical application. Since becoming a standard of The Open Group, its methodological approach relies on TOGAF [18]. TOGAF has wide audience and results from incorporation of best practices but can hardly be called a formal approach.

In the corresponding architecture framework, three enterprise layers are distinguished: business, application and technology. In comparing with DEMO, the business layer of Archimate and the ontological layer of DEMO are considered. The contents of Archimate's application and technology layers are regarded as implementation and, thus, not directly modeled in DEMO. The business layer of Archimate relates to all three B-I-D layers of DEMO, without clear distinction [19].

While Archimate does not natively address motivation, there are extension proposals [20] with this purpose. *Representing* the motivation has value in itself but, in order to reap the full benefits of addressing the motivation layer, it must go further into *engineering* the business itself. Regarding the connection with business modeling, other approaches combine Archimate and Business Modeling Canvas, a simplified version of the Business Model Ontology [21]. However, none of this approaches uses the concept of *intention*, a characteristic of the so called third wave of approaches,

like DEMO [19]. In order to fruitfully combine both approaches, the whole business layer should conform to DEMO, positively constraining the modeling activity. Enforcing the rigorously defined semantics of DEMO into Archimate's business layer is the starting point to perform Archimate-based value modeling.

**Value Modeling – e3Value.** Value Modeling was selected as it is increasingly recognized that the concept of value assists in improving communication between stakeholders of related systems, particularly Business and IT [22].

e3Value [9] is part of e3family, a set of ontological approaches for modeling 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 modeled 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.

Gordjin also introduces the concepts of *value model deconstruction* and *reconstruction* [9], comprising solution reevaluation by analysis and (re)composition of atomic components. Still, no formal decision rationale is supported and no description is provided on how to use the atomic elements beyond the original demand scope.

e3Value has also complementary approaches to modeling strategy [23] and goals [24]. In the first case, the e3forces approach positions of an enterprise in a value web. However, it is insufficient for our objectives as forces are not specified in a way that can be broken down and related with the remainder model constructs. The integration with a goal-oriented approach is very interesting but has the same limitations that GORE approaches, presented next. We acknowledge these approaches and propose to apply e3Value in a way that improves system and subsystem value modeling: inside the boundaries of organizations, as presented in section 3, instead of solely applying to e-commerce relations between formal organizations.

**Requirements Engineering – GORE.** Goal-oriented Requirements Engineering (GORE) approaches, such as I\* and KAOS (for early and late requirements, respectively) were created to trace system (requirements) to its goals, as a way to make purpose tractable. ARMOR [25] is an example of a requirements modeling language focused on motivation, that results from their combination. Despite being an advance in expressing goal structure and tracing, these approaches establish a *goal* as a consequence of having a certain strategy. However, the *goal* must not be mistaken for the strategy itself or the motivation and reasoning behind it.

In order to perform a rational and meaningful analysis it is necessary to make these forces more objective. In the Library example, *Improve Customer Experience*, *Decrease risk of non-return* or *Reduce storage space* are goals that should be decomposed and unambiguously refer to other model constructs, such as with transactions and their production facts, enforcing semantic commitment. Otherwise, they are “simply” the top nodes of a dogmatic hierarchical structure, instead of active elements system engineering. Therefore, we argue that purpose cannot be made tractable solely by applying goal-oriented requirements engineering, since known approaches fail to show the real origin of the goal as a result of upstream relations [4]. As we will see in the following section, specifying motivation in term of the needs of another system in successive engineering processes brings a greater level of objectiveness.

### 3 Towards a Solution: Principles and Framework Overview

#### 3.1 Principles of a Different Way of Thinking and Modeling

From the analysis presented in the last section, we conclude a new fundamental approach is necessary. The mindset used to devise it is using current state of the art approaches for localized modeling, with an improved teleological drive in terms of methodology and principles, presented in the remainder of this section.

**Business Service Engineering.** In most approaches, the components of the business layer can be modeled without restrictions, with the resulting incompleteness and incoherencies. For instance, Archimate's definition of Business Services lacks DEMO's B-I-D distinction. Its incorporation into Archimate is of utmost relevance to our research, since it allows 1) separating B-I-D in a layered manner and 2) raising the issue of modeling the relations between B-organizations. Therefore, we propose Business Services are validated with DEMO semantics, both in terms of completeness, by matching with the Transactional Pattern, and isolating essential (ontological – B) transactions. In turn, DEMO should relate to an e3Value model by matching actors and transactions. DEMO and e3Value have been related in [26] but not ontologically matched. A matching ontology between both is presented and discussed in [4, 27], introducing, for instance, the concepts of value and economic reciprocity to DEMO transactions and the full transactional pattern to e3Value. This allows transforming an e3Value Value Exchange into DEMO's system construction primitives.

We note that, during system development, we use e3Value without enforcing *economic independence* between actors. This restriction should only be applied in final scenarios, not for design time and exploring innovative solutions as it may leave out important business model features. For instance, in sourced activities may be taken for granted and, thus not modeled as economic independent. More importantly, transactions may not be identified at all – take for instance, *book counseling* in the Library example, which ends up being a differentiator from other offers on the market.

**Define purpose through Recursive System Engineering.** Modeling purpose as the contribution of a given system's production to its environment, as described in [4] is a very powerful teleological concept. It is important to note that the concept of *purpose* is higher in abstraction than the one of *goal* and that the later is subordinate to the former. However, *purpose* is frequently dismissed as a strategic concern and we are not aware of formal, structured, definitions of purpose that can be used as input to a system development process, neither traced back from a developed system. In this framework, purpose is as much of a strategic concern for a given system as it is a need by another. 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 elements of a system as the same concepts will recursively apply. Therefore, the *purpose* of a system (OS) must be defined in terms of the value it provides to other formal system(s) (US). In turn, this formal system will assume the role of OS towards another US.

**Model System Development Rationale.** Building on the previous principles, there is now sufficient information and conceptual tools to model each Demand (US)/Offer (OS) pair in the chain and, if relevant, to zoom in into the OS and initiate modeling of



its elements reified as independent systems. For each instance of the GSDP, the application of principles and requirements (both function and construction), driven by the needs of the US, progressively filter candidate OS's in the solution market. These primitives are, indeed, the filtering criteria that can now be specified 1) in terms of value, using a common referential and 2) relatively to adjacent elements in the value chain. This enables reasoning about the development process steps and systematically reevaluating previous assumptions and decisions.

### 3.2 Framework Overview

Our framework is based on the concept of a solution market as a means to provide purpose orientation. Systems provide services, which the customer (*using system*) gets from the provider (*object system*). Services are then valued in a market context and may be used in multiple solution chains. This way, the integration of teleological and ontological visions of a system is done by assigning market-based value semantics to each system construction element. Modeling the intermediate steps between the formulation of a need and its fulfilling by a system is, therefore, indispensable for informed and rational decisions.

In the Library example introduced in section 2, the system may be used by a customer to solve an *information need* problem, but it can also solve a *gifting* problem. According to the demand segments the system's owner wishes to address, and competing offers, different system design and engineering decisions are made. Modeling these teleological aspects is not trivial and is commonly regarded as subjective. However, by distinguishing the *contribution*, *function* and *construction* perspectives [5] business service specification can be improved.

Analyzing the Library's *loan start* transaction, we identify the value object that is functionally offered by the Library in exchange for a membership fee, not as the *book* but as its *temporary possession*. This may or not serve the interests of *using systems*, they are bound to returning the book. By defining the value of the service offered by the Library to a specific *using system* in a market context, it is possible to compare alternative solutions to procure inputs for its own value activities, i.e., *reading* or *gifting*. The *contribution perspective* abstracts any implementation choices or provisioning mechanisms. Hence, it brings the Library's production to an essential level, the first step in allowing comparison to other alternatives of bringing about such *production fact*. Some examples are internet ordering, borrowing from a friend, acquiring a digital version or even downloading an illegal copy. Each alternative introduces an offer at the solution market with specific value and dependencies, which end up providing different end-user experiences.

The combination of approaches, including DEMO, Service Science and e3Value was achieved first by theory analysis and combination, and then by ontological mapping, with modeling restrictions introduced from each theory towards the others. A formal mapping between DEMO and e3Value [9] allows enforcing the relations between perspectives: *construction* supports *function* and *function* supports *contribution*.

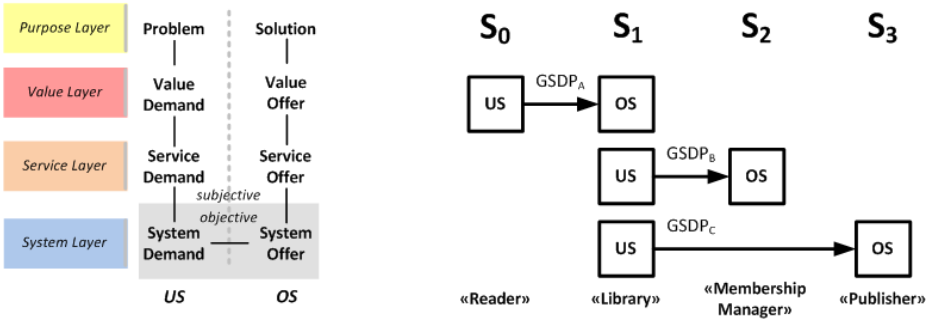


Fig. 3. Framework Overview (left) and recursive GSDP application (right)

Drawing from the principles presented in the previous section, we propose simultaneously modeling a set of layers for a given system: System, Service, Value (Market) and Purpose (Problem Solving). Their relative positioning is represented on the left part of Fig. 3. A brief description of each layer follows:

- The *System* layer defines how a service is assembled from individual elements and provided by a concrete system. Both *construction* and *function* perspectives are defined at this layer. The solution provided by a system is only effectively made operational two concrete systems, assuming the demand (US) and offer (OS) roles, connect. This concept is represented by the *objective* box in Fig. 3.
- The *Service* layer models the demands/offers a system makes/receives to/from its environment. It encapsulates a *system* as a *service system*, by providing a *partial* black-box model, framed with contract and operation conditions. Service differs from function since it incorporates the concept of *willingness* to engage into transactions. This concept is represented by a Value Port (e3Value), welcoming service interactions, as defined SD-Logic's ISPAR Model [12]. For instance, a Library could functionally provide book selling services to customers but chooses not to.
- The *Value* layer defines value exchanges composed of service offers and demands. For instance, it allows comparing a Bookstore and a Library as solutions to *provide reading content*: one fully transfers value object (*book*) rights and has unitary costs, while the other only offers temporary access at a flat rate. This layer is critical in relating the services provided by a system and a specific stakeholder, as value is always uniquely and phenomenological determined by its beneficiary [8].
- The *Purpose* layer defines the contribution of an *object system* as a solution provider to the demands of specific *using systems*. It connects service systems based on value object chaining, e.g. the Result Chain presented in [28] and, in a more narrow but formal sense, to DEMO's Result Structure [7]. For instance, the Publisher contributes by providing *book copies* which the Library can, in turn, loan to readers. Together with *service* and *value*, *purpose* fully addresses the *contribution perspective* as defined in [5] by re-scoping the development process to adjacent nodes on the value chain and recursively defining system relations (Fig. 3, right).

## 4 Conclusion

We presented a proposal for defining a system's purpose, offering a multi-disciplinary approach that integrates contributions from multiple relevant areas, resulting in a diverse conceptual foundation. Its most differentiating feature is in the way of thinking, by finding new principles that allow conceiving the relation of a system with its elements as if they are in an open market. This essentially means modeling them as separate actors from an economic viewpoint, which we argue improves modeling completeness and quality. Also, focusing on value as the bottom line means having a traceable relation between every element of the model and purpose, which assists in communication between stakeholders.

Our contribution is composed by: 1) the identification of a relevant problem space in current approaches, particularly the lack of a sound structure to model and provide an ongoing referential for *purpose*; and 2) the definition of a *conceptual framework* that addresses the issues identified in section 2 of this paper. It integrates the core concepts and their relative positioning in a layered manner, distinguishing the concepts that define a problem/solution pair end-to-end, from need to implementation.

We believe performing system modeling with a market mindset will help improving quality and change response. In this paper, we presented a combination of Enterprise Engineering, Service Science and Value Modeling. *Business* and its supporting systems can only be effectively aligned by using a sound conceptual theory, together with the corresponding methodologies and applications, integrating both the *contribution* (why), *function* (what) and *construction* (how) dimensions of a system. Formally addressing the contribution perspective of a system is critical to trace system construction to its purpose and improving Business/IT alignment.

## References

1. Laudon, K.C., Laudon, J.P.: Management Information Systems: Managing the Digital Firm. Prentice Hall
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, vol. 32(1), pp. 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: 2nd Enterprise Engineering Working Conference. Springer, Delft (2012) (forthcoming)
6. Skyttner, L.: General Systems Theory: Problems, Perspectives, Practice, 2nd edn. World Scientific Publishing Co. Pte. Ltd., Singapore (2005)
7. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer (2006)
8. Vargo, S.L., Lusch, R.F., Akaka, M.A.: Advancing Service Science with Service-Dominant Logic: Clarifications and Conceptual Development. In: Handbook of Service Science. Springer (2010)

9. Gordijn, J.: Value-based requirements Engineering: Exploring innovative e-commerce ideas. Vrije Universiteit Amsterdam, Amsterdam (2002)
10. Dietz, J.L.G.: The Atoms, Molecules and Fibers of Organizations. *Journal Data & Knowledge Engineering - Special Issue: The Language/Action Perspective Archive* 47(3) (2003)
11. Dietz, J.L.G.: Architecture - Building strategy into design. Netherlands Architecture Forum, Academic Service - SDU, The Hague, The Netherlands (2008)
12. Maglio, P.P., et al.: The service system is the basic abstraction of the service science. *Information Systems and E Business Management* (2009)
13. Pombinho, J., Tribolet, J.: Service System Design and Engineering – A value-oriented approach based on DEMO. In: 3rd International Conference on Exploring Service Science, Geneva (2012)
14. Bell, M.: Service-Oriented Modeling: Service Analysis, Design and Architecture. John Wiley & Sons, New Jersey (2008)
15. Lamsweerde, A.V.: Goal-Oriented Requirements Engineering: A Guided Tour. In: International Symposium on Requirements Engineering, Toronto (2001)
16. Regev, G., Wegmann, A.: Where do Goals Come from: the Underlying Principles of Goal-Oriented Requirements Engineering. In: 13th IEEE International Requirements Engineering Conference (RE 2005). IEEE, Paris (2005)
17. 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)
18. Lankhorst, M., van Drunen, H.: Enterprise Architecture Development and Modelling - Combining TOGAF and ArchiMate (2007)
19. Ettema, R., Dietz, J.L.G.: ArchiMate and DEMO – Mates to Date? In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO!/EOMAS 2009. LNBP, vol. 34, pp. 172–186. Springer, Heidelberg (2009)
20. Engelsman, W., et al.: Extending enterprise architecture modeling with business goals and requirements. *Enterprise Information Systems* 5, 9–36 (2011)
21. Osterwalder, A.: The Business Model Ontology - a proposition in a design science approach. Université de Lausanne (2004)
22. Cameron, B., Leaver, S., Worthington, B.: Value-Based Communication Boosts Business' Perception of IT. Forrester Research (2009)
23. Pijpers, V., Gordijn, J.: e3forces: Understanding Strategies of Networked e3value Constellations by Analyzing Environmental Forces. In: Krogstie, J., Opdahl, A.L., Sindre, G. (eds.) CAiSE 2007 and WES 2007. LNCS, vol. 4495, pp. 188–202. Springer, Heidelberg (2007)
24. Gordijn, J., Petit, M., Wieringa, R.: Understanding Business Strategies of Networked Value Constellations Using Goal and Value Modeling. In: Proceedings of the 14th IEEE International Requirements Engineering Conference. IEEE Computer Society (2006)
25. Quartel, D., et al.: A goal-oriented requirements modelling language for enterprise architecture. In: IEEE International Enterprise Distributed Object Computing Conference, Auckland, New Zealand (2009)
26. Weigand, H., Heuvel, W.-J.V.D.: A Conceptual Architecture for Pragmatic Web Services. In: First International Conference on the Pragmatic Web, Stuttgart, Germany (2006)
27. 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)
28. Boehm, B.: Value-Based Software Engineering. *SIGSOFT Software Engineering Notes* 28(2), 4 (2003)