LNBIP 146

Henderik A. Proper
David Aveiro
Khaled Gaaloul (Eds.)

Advances in Enterprise Engineering VII

Third Enterprise Engineering
Working Conference, EEWC 2013
Luxembourg, May 2013, Proceedings

Springer

Lecture Notes
in Business Information Processing 146

Henderik A. Proper
David Aveiro
Khaled Gaaloul (Eds.)

Advances in Enterprise Engineering VII

Third Enterprise Engineering
Working Conference, EEWC 2013
Luxembourg, May 13-14, 2013
Proceedings

Springer

Volume Editors

Henderik A. Proper
Public Research Centre - Henri Tudor
Luxembourg-Kirchberg, Luxembourg
E-mail: e.proper@acm.org

David Aveiro
University of Madeira
Funchal, Portugal
E-mail: daveiro@uma.pt

Khaled Gaaloul
Public Research Centre - Henri Tudor
Luxembourg-Kirchberg, Luxembourg
E-mail: khaled.gaaloul@tudor.lu

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-642-38116-4 e-ISBN 978-3-642-38117-1
DOI 10.1007/978-3-642-38117-1
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2013936542

ACM Computing Classification (1998): J.1, H.3.5, H.4-5

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

Enterprise engineering is an emerging discipline that studies enterprises from an engineering perspective. Enterprises are studied as being purposely designed and implemented systems. Enterprise engineering is rooted in both the organizational sciences and the information system sciences. The rigorous integration of these traditionally disjoint scientific areas has become possible after the recognition that communication is a form of action. The operating principle of organizations is that actors enter into and comply with commitments, and in doing so bring about the business services of the enterprise. This important insight clarifies the view that that enterprises belong to the category of social systems, i.e., its active elements (actors) are social individuals (human beings). The unifying role of human beings makes it possible to address problems in a holistic way, to achieve unity and integration in bringing about any organizational change.

Also when regarding the implementation of organizations by means of modern information technology (IT), enterprise engineering offers innovative ideas. In a similar way as the ontological model of an organization is based on atomic elements (namely, communicative acts), there is an ontological model for IT applications. Such a model is based on a small set of atomic elements, such as data elements and action elements. By constructing software in this way, the combinatorial effects (i.e., the increasing effort it takes in the course of time to bring about a particular change) in software engineering can be avoided.

The development of enterprise engineering requires the active involvement of a variety of research institutes and a tight collaboration between them. This is achieved by a continuously expanding network of universities and other institutes, called the CIAO! Network (www.ciaonetwork.org). Since 2005 this network has organized the annual CIAO! Workshop, and since 2008 its proceedings have been published as *Advances in Enterprise Engineering* in the Springer LNBIP series. From 2011 on, this workshop was replaced by the Enterprise Engineering Working Conference (EEWC). This book contains the proceedings of the third EEWC, which was held in Luxembourg.

May 2013

Henderik A. Proper
David Aveiro
Khaled Gaaloul

Enterprise Engineering – The Manifesto

Introduction

This manifesto presents the focal topics and objectives of the emerging discipline of enterprise engineering, as it is currently theorized and developed within the CIAO! Network. There is close cooperation between the CIAO! Network (www.ciaonetwork.org) and the Enterprise Engineering Institute (www.ee-institute.com) for promoting the practical application of enterprise engineering. The manifesto comprises seven postulates, which collectively constitute the *enterprise engineering paradigm* (EEP).

Motivation

The vast majority of strategic initiatives fail, meaning that enterprises are unable to gain success from their strategy. Abundant research indicates that the key reason for strategic failures is the lack of coherence and consistency among the various components of an enterprise. At the same time, the need to operate as a unified and integrated whole is becoming increasingly important. These challenges are dominantly addressed from a functional or managerial perspective, as advocated by management and organization science. Such knowledge is necessary and sufficient for managing an enterprise, but it is inadequate for bringing about changes. To do that, one needs to take a constructional or engineering perspective. Both organizations and software systems are complex and prone to entropy. This means that in the course of time, the costs of bringing about similar changes increase in a way that is known as combinatorial explosion. Regarding (automated) information systems, this has been demonstrated; regarding organizations, it is still a conjecture. Entropy can be reduced and managed effectively through modular design based on atomic elements. The people in an enterprise are collectively responsible for the operation (including management) of the enterprise. In addition, they are collectively responsible for the evolution of the enterprise (adapting to needs for change). These responsibilities can only be borne if one has appropriate knowledge of the enterprise.

Mission

Addressing the afore-mentioned challenges requires a paradigm shift. It is the mission of the discipline of enterprise engineering to develop new, appropriate theories, models, methods and other artifacts for the analysis, design, implementation, and governance of enterprises by combining (relevant parts of) management and organization science, information systems science, and computer science. The ambition is to address (all) traditional topics in said disciplines from the enterprise engineering paradigm. The result of our efforts should be theoretically rigorous and practically relevant.

Postulates

Postulate 1

In order to perform optimally and to implement changes successfully, enterprises must operate as a unified and integrated whole. *Unity* and *integration* can only be achieved through *deliberate enterprise development* (comprising design, engineering, and implementation) and *governance.*

Postulate 2

Enterprises are essentially social systems, of which the elements are human beings in their role of *social individuals*, bestowed with appropriate *authority* and bearing the corresponding *responsibility.* The *operating principle* of enterprises is that these human beings enter into and comply with *commitments* regarding the products (services) that they create (deliver). Commitments are the results of *coordination acts*, which occur in universal patterns, called *transactions.*

Note. Human beings may be supported by technical artifacts of all kinds, notably by ICT systems. Therefore, enterprises are often referred to as sociotechnical systems. However, only human beings are responsible and accountable for what the supporting technical artifacts do.

Postulate 3

There are two distinct perspectives on enterprises (as on all systems): *function* and *construction.* All other perspectives are a subdivision of one of these. Accordingly, there are two distinct kinds of models: *black-box models* and *white-box models.* White-box models are *objective*; they regard the construction of a system. Black-box models are *subjective*; they regard a function of a system. *Function is not a system property* but a relationship between the system and some stakeholder(s). Both perspectives are needed for developing enterprises.

Note. For convenience sake, we talk about the business of an enterprise when taking the function perspective of the customer, and about its *organization* when taking the construction perspective.

Postulate 4

In order to manage the complexity of a system (and to reduce and manage its entropy), one must start the constructional design of the system with its *ontological model.* This is a fully implementation-independent model of the *construction* and the *operation* of the system. Moreover, an ontological model has a *modular* structure and its elements are (ontologically) *atomic*. For enterprises the meta-model of such models is called *enterprise ontology*. For information systems the meta model is called *information system ontology*.

Note. At any moment in the lifetime of a system, there is only one ontological model, capturing its actual construction, though abstracted from its implementation. The ontological model of a system is comprehensive and concise, and extremely stable.

Postulate 5

It is an *ethical necessity* for bestowing authorities on the people in an enterprise, and having them bear the corresponding responsibility, that these people are able to *internalize* the (relevant parts of the) *ontological model* of the enterprise, and to constantly validate the correspondence of the model with the operational reality.

Note. It is a duty of enterprise engineers to provide the means to the people in an enterprise to internalize its ontological model.

Postulate 6

To ensure that an enterprise operates in compliance with its *strategic concerns*, these concerns must be transformed into generic functional and constructional *normative principles*, which guide the (re-)development of the enterprise, in addition to the applicable specific requirements. A coherent, consistent, and hierarchically ordered set of such principles for a particular class of systems is called an *architecture*. The collective architectures of an enterprise are called its *enterprise architecture*.

Note. The term "architecture" is often used (also) for a model that is the outcome of a design process, during which some architecture is applied. We do not recommend this homonymous use of the word.

Postulate 7

For achieving and maintaining unity and integration in the (re-)development and operation of an enterprise, organizational measures are needed, collectively called *governance*. The *organizational competence* to take and apply these measures on a continuous basis is called *enterprise governance*.

May 2013 Jan L.G. Dietz

Organization

EEWC 2013 was the Third Working Conference resulting from a series of successful CIAO! Workshops over the years, the EEWC 2011 and the EEWC 2012. These events were aimed at addressing the challenges that modern and complex enterprises are facing in a rapidly changing world. The participants in these events share the belief that dealing with these challenges requires rigorous and scientific solutions, focusing on the design and engineering of enterprises.

This conviction led to the idea of annually organizing an international working conference on the topic of enterprise engineering, in order to bring together all stakeholders interested in making enterprise engineering a reality. This means that not only scientists are invited, but also practitioners. Next, it also means that the conference is aimed at active participation, discussion, and exchange of ideas in order to stimulate future cooperation among the participants. This makes EEWC a working conference contributing to the further development of enterprise engineering as a mature discipline.

The organization of EEWC 2013 and the peer review of the contributions to EEWC 2013 were accomplished by an outstanding international team of experts in the fields of enterprise engineering.

Advisory Board

Jan L.G. Dietz	Delft University of Technology, The Netherlands
Antonia Albani	University of St. Gallen, Switzerland

General Chair

Henderik A. Proper	Public Research Centre - Henri Tudor, Luxembourg Radboud University Nijmegen, The Netherlands

Program Chair

David Aveiro	University of Madeira, Madeira Interactive Technologies Institute and Center for Organizational Design and Engineering - INESC INOV Lisbon, Portugal

Organizing Chair

Khaled Gaaloul	Public Research Centre - Henri Tudor, Luxembourg

Program Commitee

Bernhard Bauer	University of Augsburg, Germany
Birgit Hofreiter	Vienna University of Technology, Austria
Christian Huemer	Vienna University of Technology, Austria
Dai Senoo	Tokyo Institute of Technology, Japan
Eduard Babkin	Higher School of Economics, Nizhny Novgorod, Russia
Emmanuel Hostria	Rockwell Automation, USA
Eric Dubois	Public Research Centre - Henri Tudor, Luxembourg
Florian Matthes	Technical University of Munich, Germany
Gil Regev	École Polytechnique Fédérale de Lausanne (EPFL), Itecor, Switzerland
Graham McLeod	University of Cape Town, South Africa
Hans Mulder	University of Antwerp, Belgium
Jan Hoogervorst	Sogeti Netherlands, The Netherlands
Jan Verelst	University of Antwerp, Belgium
Joaquim Filipe	School of Technology of Setúbal, Portugal
Jorge Sanz	IBM Research at Almaden, California, USA
José Tribolet	INESC and Technical University of Lisbon, Portugal
Joseph Barjis	Delft University of Technology, The Netherlands
Junichi Iijima	Tokyo Institute of Technology, Japan
Marielba Zacarias	University of Algarve, Portugal
Martin Op 't Land	Capgemini, The Netherlands Antwerp Management School, Belgium
Natalia Aseeva	Higher School of Economics, Nizhny Novgorod, Russia
Olga Khvostova	Higher School of Economics, Nizhny Novgorod, Russia
Paul Johanesson	Stockholm University, Sweden
Peter Loos	University of Saarland, Germany
Pnina Soffer	Haifa University, Israel
Remigijus Gustas	Karlstad University, Sweden
Robert Lagerström	KTH - Royal Institute of Technology, Sweden
Robert Winter	University of St. Gallen, Switzerland

Rony Flatscher	Vienna University of Economics and Business Administration, Austria
Sanetake Nagayoshi	Tokyo Institute of Technology, Japan
Stijn Hoppenbrouwers	HAN University of Applied Sciences, The Netherlands
Ulrich Frank	University of Duisburg-Essen, Germany

Table of Contents

Value Orientation

Value-Oriented Solution Development Process: Uncovering the Rationale behind Organization Components 1
João Pombinho, David Aveiro, and José Tribolet

Enterprise Change

Towards Developing a Model-Based Decision Support Method for Enterprise Restructuring 17
Eduard Babkin and Alexey Sergeev

Exploring Organizational Implementation Fundamentals 28
Martin Op 't Land and Marien Krouwel

A Case Study on Enterprise Transformation in a Medium-Size Japanese IT Service Provider: Business Process Change from the Ontological Perspective 43
Sanetake Nagayoshi

Explaining with Mechanisms and Its Impact on Organisational Diagnosis 58
Roland Ettema, Federica Russo, and Philip Huysmans

Transformation of Multi-level Systems – Theoretical Grounding and Consequences for Enterprise Architecture Management 73
Ralf Abraham, José Tribolet, and Robert Winter

Requirements Engineering and Entropy issues

Identifying Combinatorial Effects in Requirements Engineering 88
Jan Verelst, Alberto Rodrigues Silva, Herwig Mannaert, David Almeida Ferreira, and Philip Huysmans

Understanding Entropy Generation during the Execution of Business Process Instantiations: An Illustration from Cost Accounting 103
Peter De Bruyn, Philip Huysmans, Herwig Mannaert, and Jan Verelst

Author Index 119

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).

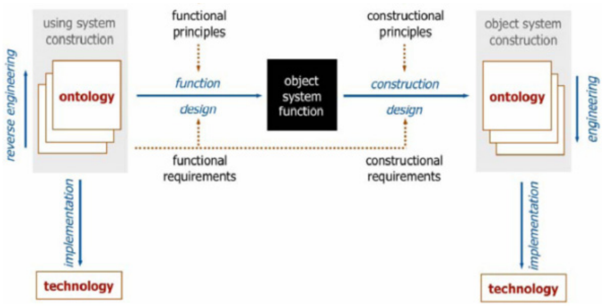

Fig. 1. Generic System Development Process [6]

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.*

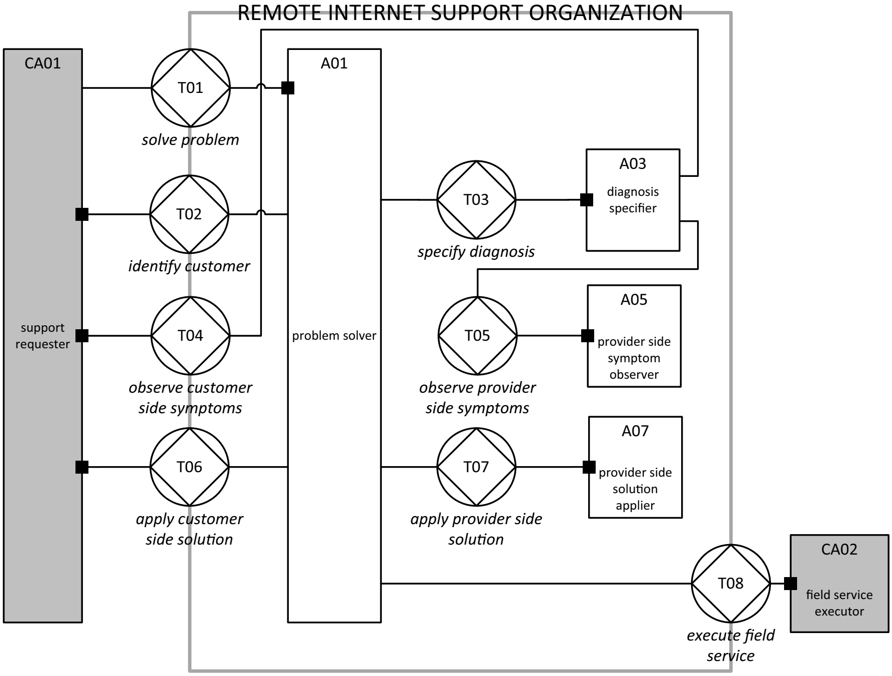

Fig. 2. Unimedia Remote Internet Support Actor Transaction Diagram (ATD)

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

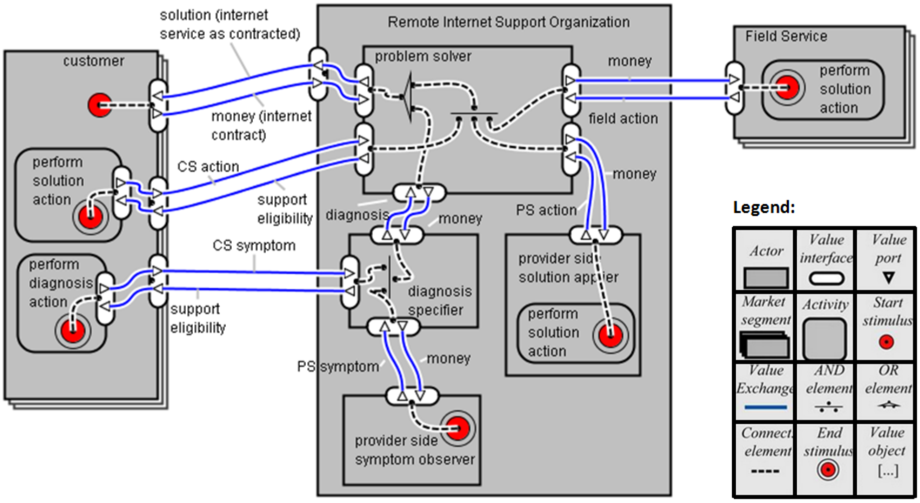

Fig. 3. Value model for Remote Internet Support scenario

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.

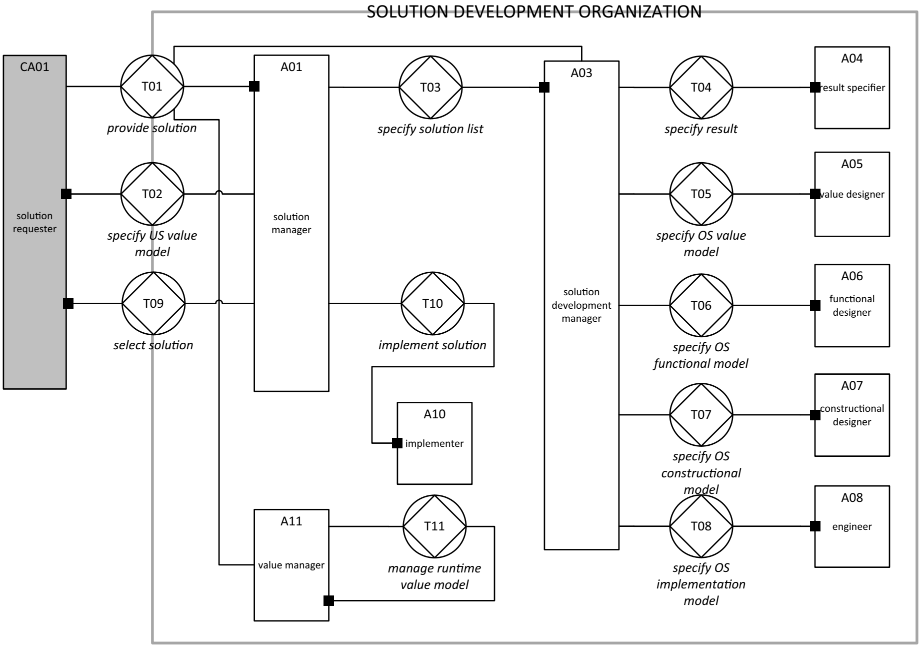

Fig. 4. Solution Development Organization – ATD

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