Aligning e3Value and DEMO – Combining Business Modelling and Enterprise Engineering

João Pombinho\textsuperscript{1,2}, David Aveiro\textsuperscript{3}, José Tribolet\textsuperscript{1,2},

\textsuperscript{1} INESC-ID, INOV, Rua Alves Redol 9, Lisbon, Portugal
\textsuperscript{2} Department of Information Systems and Computer Science, Instituto Superior Técnico
Technical University of Lisbon, Portugal
\textsuperscript{3} Exact Sciences and Engineering Centre, University of Madeira, Funchal, Madeira, Portugal

jpmp@tecnico.ulisboa.pt, daveiro@uma.pt, jose.tribolet@inesc.pt

Abstract. In this paper, we present a match between two ontologies, DEMO and e3Value, with the aim of bridging Business Modelling and Enterprise Engineering by providing an objective set of concepts for integrating the teleological and ontological perspectives of a system. DEMO contributes to the constructability of a business model, by providing a theory and method for designing and engineering enterprises. e3Value contributes to the justifiability of a given system construction, providing the notion of purpose through value semantics, network context and economic concepts, such as reciprocity. The resulting ontology includes a formal definition of shared core concepts and introduces positive constraints which, we argue, improve system model quality by combining two relevant and complementary approaches.

Keywords: Business Modelling; e3Value; DEMO; Enterprise Engineering; Alignment.

1 Introduction and Motivation

The need of consistently and coherently design organizations is an increasingly pressing issue to modern enterprises. In fact, both academia and industry seem to have definitively embraced the topic, and rightfully so. 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, misalignments between the business and its support systems are frequently appointed as a reason of failure [2]. 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.

This paper follows-up on [3] and presents the result of an ontology integration between DEMO [4], from the perspective of Enterprise Engineering, and e3Value [5], from Value Modelling. A method for value-oriented system development has also been proposed [6], providing a specific context for the ontology specification here.
presented. In this sense, as it is used as a means to pursue teleological and ontological integration in a system development context, it fundamentally differs from other approaches, such as [7].

DEMO contributes to the constructability of a business model, by providing a theory and method for designing and engineering enterprises. e3Value contributes to the justifiability of a given system construction, providing the notion of purpose through value semantics, network context and economic concepts, namely reciprocity.

2 Matching e3Value and DEMO Ontologies

In this section we present a formal alignment ontology specification (cf. Fig. 1) and, following, the main concepts and the rationale for their matching in our proposal.

For illustration purposes, we will use the classic Library case from Enterprise Ontology [4], instantiated in a value model and a constructional model (cf. Fig. 2). The models are deliberately assumed to have been created independently.
Value object exchanges define the motivation for establishing relations between actors. Value networks form by acting upon the desire of a value object by a given actor. The successive bonding between actors forms a chain of value objects exchanged as parts of a solution from a given actor’s perspective. For instance, to satisfy the desire of reading a physical book, a chain of value objects forms to deliver the rights to a particular book copy. The network is a combination of these partial perspectives (i.e., from individual actors, such as reader, library, stock manager, publisher) and a cumulative of chains representing (possibly) indirect dependency.

Figure 2. Simplified e3Value model and DEMO construction model for Library example

**Value Object.** The first step is to define which value objects are to be exchanged and, more importantly, why. We find an important relation between production fact and value object classes since the production of each transaction determines its effective contribution to the value chains it participates in. Hence, a production fact must be about a certain value object, otherwise it is meaningless. Conversely, a value object that is not referred in a production fact effectively does not exist in the overall model.

**Example:** investigating what is the value brought about by the stock controller yields two different contributions. The first contribution is title availability management and consists in combining title demand information with the current stock (considering the non-returned books) to minimize the number of declines of the loan book transaction due to unavailability of the title. The second contribution is actually ordering the titles that were identified as being on-demand efficiently. To model these contributions, two new value objects were identified: book demand and budget (cf. Fig. 3). Integrating these two value objects in the model implies the identification of additional constructs, namely actors and transactions, as presented next.

**Actor.** Actors are the active elements of both social systems and value networks. In DEMO, an actor is a subject fulfilling an actor role in a transaction type. The initiator and executor actor roles of a given transaction are bound by their common interest in bringing about a production result. In e3Value, both actors (producer and consumer) are bound by the willingness to share value objects with the concept of reciprocity.
A particular subject instantiating the actor role sets the transactional valuations of the economic actor (also including investment and expenses). Then, in addition to the structure of the actor role, it is the instantiation with a subject that allows fully specifying the economic actor. The possible differences between these two may determine the viability of the design. This important distinction allows defining the economic actor from e3Value as a composite of both actor role and subject concepts of DEMO.

**Example:** The matching ontology prompts for the definition of the economic actor in the value model that corresponds to the *stock controller* actor (A09) in the ontological model. Besides the actor role that is defined by the ontological transaction it takes part in (*manage stock*), A09 has a value interface and performs a value activity, let us name it *procure book*. Both are now made clear as separate actor roles in Fig. 3.

**Value Transaction.** In DEMO, a transaction is a sequence of acts following a specific pattern between two actors. One of the actors assumes the role of *initiator* and the other assumes the role of *executor*. For instance, in requesting a book, the Book Reader is the initiator and the Library is the executor.

A *value interface* belongs to a given actor and is composed by a set of value ports (at least one in each direction). The concept of connecting value ports is found in e3Value as a *value exchange*. However, it is not an exchange in fact, but only a flow (or transmission) of exactly one value object. In this concern, we propose the concept of *value transmission* as a better notion to express the flow of a value object from one actor to another. In a *value transmission*, the initiator has an inbound value port and the executor has an outbound value port. A minimum of two value transmissions in each direction of a value interface constitutes a *value transaction*. This specification more clearly enforces the notion of *economic reciprocity*.

**Example:** In order to model exactly how the *stock manager* participates in the value chain, its value interfaces and value activities must be specified. In this case, action rules would specify the operational dependencies and order of execution, for instance between making *budget* available to the buyer and effectively ordering the *book*.

**Process.** The original intention for a given value transaction can generally be implemented in many ways. Some will honour the original intention and some will not. For instance, e3Value patterns such as monitoring have similar value model representations and rely on process implementation to achieve the original intention [8]. In our proposal, both value transaction and value interface are implemented by action rules specification. In this way, the grouping of value transmissions (in the case of value transactions) and the grouping and dependencies of value ports (in the case of value interface) are formally defined in DEMO through conditional links and response links present in action rules. Therefore, we say that DEMO’s action rules realize both value transactions and value interfaces, as action rules will specify the conditions that must be complied to in order to execute a certain chain of transactions.

**Competence.** In DEMO, the concept of competence is rigorously defined as “the ability of a subject to perform a particular type of production act as well as the corresponding coordination acts”. A *value port* in e3Value represents the offer or request
of a certain value object to or from the environment. We specified one class to accommodate the notion of inbound value port and another one for the notion of outbound value port as we can now find a direct correspondence, for example, between the outbound value port and the transaction execution class: only an actor that is competent to perform the promise and state c-acts and the p-act will be allowed to be executor of the respective transaction and, thus, of offering a specific value object.

It is now possible to specify the necessary competences a subject to implement a value actor by identifying the individual acts and facts pertaining its role in the transaction (initiator or executor)

**Example:** book procurer (initiator) must be competent to accept or reject a budget stated by the stock manager (executor).

### Aligning Value and Construction Models

As a result of applying the ontology to align a pair of value and construction models, a revision of the initial Library model was produced:

![Figure 3. Library e3Value Model (revised)](image)

Using the concepts from the matching ontology, a set of core rules was defined in order to be able to identify misalignment points between an e3Value Model and a DEMO ontological model. These rules are presented as model alignment qualities:

1. **Value Object completeness** – for each actor, if a VO is exchanged in outbound value exchange, then it must either be exchanged in inbound value exchange or be generated inside, by a value activity; for instance, in the Library example, is the Book value object traceable from the final customer to its origin?
2. **Transaction completeness** – the acts and facts of the complete transactional pattern are fully specified; for instance, what happens if books are not returned, i.e., no state of the return book transaction is made?
3. **Value exchange constructability** – for each value exchange, the corresponding ontological transaction 1) exists and 2) is fully specified (2);
4. **Transaction justifiability** – for each ontological transaction, the corresponding value exchange 1) exists and 2) is complete (1);
5. **Economical reciprocity** – the dependencies between the ontological transactions that constitute a value transaction are defined via Action Rules;
6. **Valuation** – for each actor, the valuation of value ports, the Investment and the Expenses are defined;

7. **Implementability** – there is an actor to instantiate each actor role (value port structure) according to its valuation specification.

### 3 Contributions and Conclusion

In this paper we have highlighted advances regarding Business Modelling and Enterprise Engineering integration state of the art. Our proposal includes:

- bidirectional alignment, as it is possible not only to begin with a value model and check if a construction model is consistent, but also the other way around;
- formal specification of the integration ontology that unambiguously relates an e3Value and DEMO models;
- detailed matching in order to take advantage specific semantics from both sides, such as competences and action rules specification;
- a core set of rules that can be used to check model quality and coherence.

Our aim is to provide an objective set of concepts for integrating the teleological and ontological perspectives of a system. DEMO contributes to the constructability of a business model by providing detailed transactional modelling. e3Value contributes to the justifiability of a given system construction, providing the notion of purpose by explicating the network of value objects generated by each production fact as transaction results. Moreover, e3Value provides network context which, in turn, allows positioning and relating DEMO models across a given value chain.

There are still limitations in addressing the specification of value chains as there is no direct correspondence in DEMO so far. The extension of our ontology to incorporate connectors and stimuli is currently ongoing and will be presented as future work.

A method for Value-oriented System Development [6] using the presented ontology has been defined [6] and has been used in practice, as reported in [9].

### 4 References


