Toward Semantics-aware Representation of Digital Business Processes

Lianne Bodenstaff

Information Systems Group, Dept. of Computer Science, University of Twente
E-mail: l.bodenstaff@utwente.nl

Paolo Ceravolo, Cristiano Fugazza

Dept. of Information Technologies, Università degli Studi, Milan
E-mail: {ceravolo, fugazza}@dti.unimi.it

Andreas Wombacher

School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne
E-mail: andreas.wombacher@epfl.ch

An extended enterprise (EE) can be described by a set of models each representing a specific aspect of the EE. Aspects can for example be the process flow or the value description. However, different models are done by different people, which may use different terminology, which prevents relating the models. Therefore, we propose a framework consisting of process flow and value aspects and in addition a static domain model with structural and relational components. Further, we outline the usage of the static domain model to enable relating the different aspects.

Keywords: Business Process Modeling; Business Rules; Value Modeling; Consistency Checking.

1. Introduction

The term *extended enterprise* is meant to designate any collection of organizations sharing a common set of goals. An enterprise can be a whole corporation, a government organization, or a network of geographically distributed entities. It is fundamental to EE applications to support digitization of traditional business processes, adding new processes enabled by e-business technologies (e.g. large scale Customer Relationship Management). In this scenario, *Business Process Modeling* (BPM) techniques are becoming increasingly important. BPM is used to design the orchestration mechanisms driving the interaction of complex systems, including communication with processes defined and executed by third parties. OMG's *Model Driven Architecture* (MDA) (1) provides a framework for representing processes at different levels of abstraction. In this paper, we rely on a MDA-driven notion of business process model, constituted by three distinct components: A *static domain model*, including the domain entities (actors, resources, etc.), a *workflow model*, providing a specification of process activities, and a *value model*, describing the value exchange between parties. Section 2 introduces the four-layer model resulting from the combination of these distinct aspects of BPM. In the modeling of static models, covered in Section 3, we shall focus on expressive formalisms constituted by controlled fragments of natural languages, introducing their translation into logic-based static domain models, and describing their relations with Semantic Web (SW) metadata formats (17). The static model also provides a comprehensive description of the entities that interact with each other in the *workflow model* and the resources that are exchanged during workflow execution, as explained in Sec. 4. Logic-based models can be easily derived from workflow descriptions and can therefore be integrated with the static model for checking consistency and computing business metrics. In Sec. 5, the overall model is completed by providing the *value model* underlying business processes. As for process models, these are also typically expressed by means of visual languages and can be translated into logic-based representations to obtain data structures that are amenable to automated processing. Section 6 outlines a simple example of conjunct evaluation of all layers in the model. Finally, Sec. 7 draws the conclusions. The contribution of the paper is the introduction of the framework and the outline of a mapping of the different layers into a logic based representation enabling reasoning on the properties of a system described using the framework. The framework is illustrated on a pay-per-view business case as it can be found in many web pages right now (comparable to e.g. dossiers in www.spiegel.de). Since credit card payment has high transaction costs, several credentials (like e.g. electronic coins) are provided by a single transaction which can be spent for viewing content afterwards.

2. Modeling business domains

The high expressivity that is required by rules has led business analysts toward the adoption of natural language as the encoding formalism for *business rules* (BR). This strategy clearly fulfils the need of knowledge sharing between humans, but inevitably complicates any sort of automated processing on rules. A tradeoff between expressivity and formal specification of statements is constituted by *controlled natural languages* and, among them, *controlled English* (CE): these formalizations are derived from natural languages by constraining the admissible sentential forms to a subset that is both unambiguous and expressive. Fig. 1 displays the four layers composing our framework. Top to bottom, the first two layers represent distinct facets of the static domain model and require different inference techniques. They are defined by using CE business rules and are partially translated into logic-based formalisms. Respectively, the *structural* component is constituted by OWL ontologies (18) while the *relational* component is made of more general SWRL rules (3). The conjunct evaluation of these data layers represents one of the main challenges of reasoning on BR models, as explained in Sec. 3. Another important component of a successful business strategy is related to the organization of process workflows. To this purpose, a business process is viewed as the sequence of activities and decisions arranged with the purpose of delivering a service, assuring security and effectiveness, in accordance to the service life cycle. The third layer in Fig.1 builds upon the entities introduced in the topmost layers and articulates process workflows by using a visual formalism. It allows for a one-to-one translation into the logic-based formalisms envisaged by the static domain model. Finally, a model especially important for describing inter-organizational collaborations is a value model, estimating profitability for every actor involved in the collaboration. A value model can also be translated into either OWL or SWRL entities. The representation of the different facets in a logic based representation facilitates checking of properties overlapping the dimensions of a single model. A basic property is the contradiction freeness of the representations of the different facets.

Fig. 1. The overall model.

2.1. *Related work*

Representing a system in different facets is not new and there exist plenty of different approaches (e.g. GRAAL or Zachman framework). Most of these approaches are high level and do not provide an operationalization of property checking. The approach closest to the one described in this paper focusses on business flow, value model, and implementation of a system (20). Checking of static and dynamic consistency (19,21) is provided on a non logic-based abstraction. For each facet there is plenty of work on how to represent a particular facet and to check properties within a single facet. With regard to CE formalisms, a widely acknowledged example is the *Attempto Controlled English* (ACE) (10), a general-purpose controlled natural language supporting specification of complex data structures, such as ontologies. General-purpose controlled languages can be provided with a formal (e.g. logic-based) semantics; however, they fall short of being capable to model all the aspects of a business domain. With regard to the expressive power required by BR, rule-based languages may need to cover higher order logics and also may specify the modal interpretation to be associated with a statement. The recognition of these requirements was a major driver of OMG's *Semantics of Business Vocabulary and Business Rules* (SBVR) proposal (14), aimed at specifying a business semantics definition layer on top of its software-oriented layers. The reader can refer to (7) for a more complete survey of controlled natural language formalisms. With regard to workflow languages, a broad range of standards allowing formalization of process flows exist e.g. using event based approaches (e.g. StateCharts), or logic-based approaches (e.g. transaction logic (5)). Dependent on the formalism operations are provided to check properties within a single model or between different models. However, these formalisms are hardly used in non academic environments, therefore, we use OMG's Business Process Modeling Notation (BPMN) (2), which was designed as a tradeoff between simplicity of notation and expressiveness. A value model facilitates a cost-benefit analysis of a business scenario based on e.g. Net Present Value. Models supporting this analysis are e.g. REA and Business Modelling Ontology as well as e^3-value model (22), which will be used further on due to its graphical representation.

3. Rule-based Structural Description

By using CE formalisms for expressing BR, it is possible to apply translation mechanisms that lead to an univocal logic formulation of rules. Consider the following ACE statement:

A customer provides a credit card to a retailer. (1)

This rule can be translated into Discourse Representation Structures (DRS) (9) that represent a subset of FOL and provide a pathway to executable logic formulations [a]. A subset of ACE can be mapped with the OWL DL ontology language (18); it can therefore take advantage of DL *reasoners* (11,15) to infer implied knowledge. Unfortunately, DL represents only a small fragment of FOL; particularly, it is also limited to expressing binary relations between entities. As a consequence of this, even the simple ternary relation binding *customers*, *retailers*, and *credit cards* in (1) cannot be expressed with OWL built-in constructs. For a more traditional processing of ACE rules, DRS can also be translated into RuleML (4) to be processed by *rule engines*, such as the Jena framework (12). Note that, in this case, the term "rule" is not indicating a BR, but instead the Horn fragment of FOL which guarantees a sound and relationscomplete reasoning on rules by applying either forward- or backward-chaining derivations. OWL DL ontologies and SWRL rules are associated with different inference paradigms (i.e., the capability of deriving implied knowledge). They both are necessary in the evaluation of business process models in the EE scenario. Specifically, information under full control of the stakeholder (e.g., a company) writing the model (e.g., the notion of *employee*) can be modeled as in traditional database design. In this case, BR simply provide a lingua franca by means of which business analysts and software developers can more easily translate company data requirements into real-world implementations. Other knowledge, however, needs to be introduced in order to compete and cooperate in the EE scenario (e.g., the notion of *competitor*); this knowledge is not under the modeler's full control, and may therefore be incomplete. This kind of incomplete descriptions may also express proprietary entities from within the business model. In fact, the complexity of business descriptions that need to be expressed by BR may not make it possible to exhaustively express the business domain. We indicate by *closed-world assumption* (CWA) the approach implemented by applications that only process complete data under the full control of their owner, expressed by the relational data layer in Fig. 1. In this case, failing to retrieve answers to a query (say, '*retrieve the credit card data associated with customer John Smith*') automatically implies that such data do not exist. Consequently, customer John Smith constitutes a valid answer to the query '*retrieve customers that do **not** have a credit card associated with them*' because incomplete knowledge amounts to false (i.e., negative) knowledge. This notion of negation (generally referred to as *negation as failure*) leads to the *non-monotonic reasoning* that provides the correct interpretation of closed systems. In the context of logic inference, the term 'non-monotonic' essentially means that conclusions (e.g., that *John Smith* is a valid answer to the previously defined query) may be contradicted by adding information to the knowledge base (e.g., the assertion '*customer John Smith provides the credit card VISA-041*'). On the contrary, we indicate by *open-world assumption* (OWA) the monotonic approach to inference that should be applied to heterogeneous data sources, such as those collected by individual systems in the EE scenario, and also (according to business analysts) to proprietary descriptions expressed by business rules, wher-

[a] Recall that full FOL is proven to be undecidable; therefore, deriving the DRS corresponding to an ACE statement does not imply that such logic formulation can also be executed by programs.

ever not explicitly stated otherwise. OWA represents a fundamental requirement for Semantic Web (SW) languages (17) and, among them, OWL ontologies. The structural component of the knowledge base may express data structures that cannot be considered as complete knowledge.

4. Declarative Process Flow Description

Graphical process flow models support immediate reading by all people involved like e.g. business analysts designing a process, or technical developers implementing it. Further, the graphical model can be used as a basis for formalization of the process flow to enable checking of model properties, specifying model exchange formats, and annotating models. In this paper, we use BPMN as the graphical notation to introduce the example depicted in Fig. 2. The example is an inter-organizational workflow containing several stakeholders, each represented as a *pool*. The process model consists of tasks represented as rectangles describing a unit of work and decision points represented by diamonds. These constructs are connected with each other by a control flow (causal relationship) represented by solid arrows and a data flow (exchange of messages) represented by dashed arrows. The example process involves three stakeholders *Target Market* representing all clients, *Content Provider*, and *Credit Institution*. The target market initiates the process by requesting a content, which is answered by the Content Provider issuing a request for credentials. When the Target Market wants to buy credentials (request credentials message), the Content Provider creates an invoice and the Target Market provides the payment information. The Content Provider can now perform a payment process with the Credit Institution resulting in a successful or denied payment. In the first case, the content acquisition process continues; in the second case, the process is terminated. When the Target Market gives credentials for the content, the Content Provider checks the validity of the credentials and provides the content to the Target Market.

Based on the graphical notation, a formal representation of the BPMN model is provided for *Consistency Checking*. Here, declarative formalizations play the main role. Traditional formalisms aim at verifying execution properties of workflow models. For instance, a typical problem is to identify if a path is terminating or which tasks are in dependency with others. Declarative formalizations cannot support this kind of controls but act very well for evaluating the consistency of the objects acting in the transaction or the data objects exchanged in the transaction, as discussed in (11).

5. Declarative Value Model Description

Value models are means to perform a cost-benefit analysis as a basis to decide on the profitability of a business. To discuss different scenarios and to explicate dependencies between different actors, a graphical representation is again of advantage. In this paper we use e^3-value notation (22) to introduce the example illustrated in Fig. 3. The example consists of two actors, that is the *Content Provider* and the *Target Market*. The Target Market has a customer demand of getting access to content, which can be accomplished in two different ways (OR-split). Either the Target Market possess credentials then these credentials can be exchanged for content, or the Target Market first buys credentials and exchanges some of them for content. The value model, in addition, contains estimates on the number of customer demands, the number of customers in the Target Market, the price for a single credential and the price for the content. Based on this information profitability of the business scenario can be evaluated. Based on the graphical representation of the value model, a formal representation will be provided to enable model property checking, model exchange format definition, and annotation of value models.

Property Checking. Properties of value models focus on profitability, which is based on doing calculations using estimates provided by the modeler. Declarative representations do not seem appropriate for these kinds of properties, but provide the good means to evaluate the consistency of objects acting in the business scenario and data objects exchanged between the different actors. The aim is to enable a semi automatic checking of object relations compared to the manual checking in (19).

Model Exchange Formats. Similar to the process flow model, a declarative description of a value model is a notation independent formalism and therefore is a good basis for transforming one notation into another one as long as the expressiveness of the formalisms is equal.

Annotation. In addition to the collaborative devel-

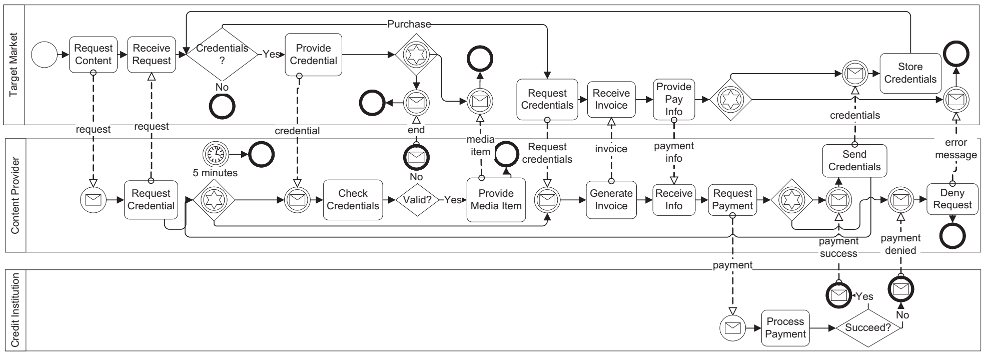

Fig. 2. Pay per view Process model.

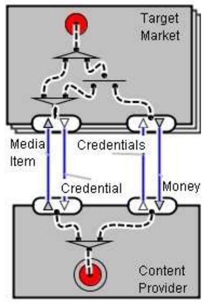

Fig. 3. Pay per view Value model.

opment of a value model (in accordance with the process flow section) they can also be used to monitor the progress of a business as e.g. described in (21). Here annotations are very helpful to correlate the manual observations of the business development to the value model.

6. Conjunct Evaluation of Models

The primary advantage of integrating into a common framework the distinct aspects introduced so far is cross-checking of constraints on the basis of the shared vocabulary expressed by the static domain model. Specifically, this Section exemplifies the constraints posed by the value model on the workflow. Apparently, the Content Provider's pool in Fig. 2 could also be seen as the combination of two independent processes. In fact, the first part of the workflow regulates the fruition of media items by customers on the basis of credentials exchange. The corresponding implementation will amount to checking credentials w.r.t. the cryptographic scheme adopted by the content provider. Instead, in the second part the content provider sells credentials (e.g., virtual coins) to customers by relying on a credit institution that takes care of the e-commerce transaction and, once a transaction is successfully completed, sends a notification to the content provider. The two parts are essentially independent inasmuch their time-spans do not necessarily overlap (virtual coins may be spent at any time past their purchase) and also they may involve different individuals (virtual coins can be further distributed to third parties). Nevertheless, implementing these services as separated components may fail to portray some important aspects of the collaboration between the Content Provider and the Target Market. As an example, Fig. 3 clearly draws the correspondences between entity Money, representing revenue, and entity MediaItem, the delivered media product[b]. Consequently, when validating the workflow model against the value model, it is necessary that this correspondence also holds: in Fig. 2, this relation is expressed by the control flow linking activity Store Credentials (in the right-hand side of the Target Market's pool) with the decision point in the beginning of the process. Should the individual subprocesses (i.e., buying credentials and accessing media items) be kept separated, with no link between credential purchase and media item fruition, the violation would be reported. This kind of a priori consistency check is enabled by the shared logic foundations of the two models. The integrated models can also be used to ground the a posteriori checking of violations, such as a clever hacker succeeding in producing a forged credential that is considered valid by the media items' delivery system. In fact, this would constitute a violation to the regulations posed by the value model, i.e. that a valid credential also has to match the correponding payment information.

7. Conclusions

An extended enterprise can be described with different models, each focusing on a particular aspect. A comparable approach in software engineering is Model Driven Architecture. The challenge having several models describing a single enterprise is to enable relating the different models with each other. This is not straightforward since different models are

[b]This correspondence is drawn by means of concept Credentials, that is defined as aggregation of concept Credential

done by different people, who seldom use the same terminology and tools. We introduced a framework covering process flow and value models as the aspects of an enterprise and support relating them by introducing an additional layer: the static domain model consisting of structural and relational components. The aspects and the static domain model directly facilitate to check object consistency, i.e., whether the terms and their usage are consistent with the static domain model, to transform aspects from one formalism to another one, and to support annotating the models. Further on, the checking of relations between different aspects has to relate the different objects in the models and therefore can benefit from the formal representation of the static domain model. In future work, we will elaborate on the formalization of the process flow and the value model as well as further evaluate our ideas by doing case studies.

Acknowledgments

This work was partly funded by the Italian Ministry of Research under FIRB contract n. `RBNE05FKZ2_004`, `TEKNE` and by the Netherlands Organisation for Scientific Research (NWO) under contract number 612.063.409.

References

1. Aßmann, U., Aksit, M., and Rensink, A., editors (2005). Model Driven Architecture, European MDA Workshops: Foundations and Applications, MDAFA 2003 and MDAFA 2004.
2. Bauer, B., Müller, J., and Roser, S. A model-driven approach to designing cross-enterprise business processes. In *OTM Conference Workshops*, volume 3292/2004 of *LNCS*.
3. Boley, H., Dean, M., Grosof, B., Horrocks, I., Patel-Schneider, P. F., and Tabet, S. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML.
4. Boley, H., Tabet, S., and Wagner, G. (2001). Design rationale
of RuleML: A markup language for semantic web rules. citeseer.ist.psu.edu/boley01design.html.
5. Bonner, A. J., Kifer, M. (1998). A Logic for Programming Database Transactions. Logics for Databases and Information Systems, pages 117–166. Kluwer Academic Publ.
6. Ceravolo, P., Damiani, E., and Viviani, M. (2007). Bottom-up extraction and trust-based refinement of ontology metadata. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):149–163.
7. Controlled Natural Languages. http://www.ics.mq.edu.au/r̃olfs/controlled-natural-languages/.
8. Fox, J. and Borenstein, J. (2005). XMI and the many metamodels of enterprise metadata. In *XML conference and exhibition*.
9. Fuchs, N. E. and Schwertel, U. (2003). Reasoning in Attempto Controlled English. In *Workshop on Principles and Practice of Semantic Web Reasoning (PPSWR 2003)*, Lecture Notes in Computer Science, Hannover. Springer.
10. Fuchs, N. E., Schwertel, U., and Schwitter, R. (1999). Attempto Controlled English (ACE) Language Manual, Version 3.0. http://attempto.ifi.unizh.ch.
11. Haarslev, V. and Moller, R. (2001). Description of the RACER system and its applications. In *International Workshop on Description Logics*.
12. Jena, A Semantic Web Framework for Java. http://jena.sourceforge.net/.
13. Melnik, S. and Decker, S. (2000). A Layered Approach to Information Modeling and Interoperability on the Web. In *Semantic Web Workshop*.
14. OMG (2006). Semantics of Business Vocabulary and Busi-
ness Rules Specification. http://www.omg.org/cgi-bin/apps/doc?dtc/06-08-05.pdf.
15. Parsia, B., Sivrin, E., Grove, M., and Alford, R. (2003). Pellet OWL Reasoner. Maryland Information and Networks Dynamics Lab, http://www.mindswap.org/2003/pellet/.
16. van der Aalst, W., Hofstede, A., and Weske, M. Business process management: A survey. In *Int. Conf. on Business Process Management (BPM 2003)*, volume 2678/2003 of *LNCS*.
17. W3C. Semantic web activity. http://www.w3.org/2001/sw/.
18. W3C (2004). OWL Web Ontology Language Overview. http://www.w3.org/TR/owl-features/.
19. Zlatev, Z. and Wombacher, A. (2005). Consistency between e^3-value models and activity diagrams in a multi-perspective development method. In *OTM Conferences (1)*, pages 520–538.
20. Bodenstaff, L., Reichert, M., Wieringa, R. (2007). Towards the Integration of Value and Coordination Models In Workshop on Business Process Modeling, Development, and Support (BPMDS).
21. Bodenstaff, L., Wombacher, A., Reichert, R., Wieringa, R. (2007). Monitoring Collaboration from a Value Perspective In IEEE Intl. Conf. on Digital Ecosystems and Technologies (DEST).
22. Gordijn, J., Akkermans, J. M. (2003). Value-based Requirements Engineering: Exploring Innovative e-Commerce Ideas Requirements Engineering, 8(2):114–134.