Setting up an ontology of business models

Alexander Osterwalder, Christine Parent, Yves Pigneur
Université de Lausanne

Introduction

The overall goal of the ongoing research described in this paper is to combine business management research in the domain of business models with more technical and conceptual research in the domain of informatics, particularly ontologies. In other words we aim at applying the rigor, precision, more complete descriptions (e.g. constraints and rules) and reasoning of ontologies to the concept of business models. The formalization of a business model ontology will provide the conceptual foundation for new methods and computer-based tools for such diverse fields as management-level business model design, business strategy & Information Technology/Information Systems alignment, more technical process & requirements engineering, and automatic comparison of business models . This more formal approach will remove eventually existing ambiguities and will allow the use of the reasoning capabilities of an ontology based upon a logic language to check the consistency and satisfiability of the business model, and complement the business model with integrity constraints and deduction rules.

A business model reference model (i.e. business model ontology) as proposed in [Osterwalder 04] and outlined in the next section is a first step on the way to clarifying what terms and concepts belong into a business model and how they relate to each other. However, the proposed model is not yet sufficiently rigid and formal for building the foundation for more sophisticated requirement elicitation methods and computer-based tools. Section 3 shortly sketches the research in progress in order to formally model the ontology we have designed. Section 4 shortly compares two extreme kinds of models, description logics and conceptual models, that could be selected, and gives the rationale for adopting an OWL paradigm for defining our ontology. The last section presents our claim that modeling the business model of companies, and not only the enterprise model, should contribute to improving interoperability.

A reference business model (= one relevant recent result)

Our motivation to design a business model ontology stems from the fact that the power of computer-based design tools has had little influence on the science of management to date although defining the business logic or the structure of a company has a very strong design component. This stands in strong opposition with other fields such as such as engineering where design is a natural component and Computer Aided Design (CAD) has become irreplaceable. Opponents to applying computers to management science will stress the complexity of a business enterprise and the importance of the role of human judgment. We don't deny this but simply think that computer assistance can bring a fresh breeze to business model design and management. Particularly, since Information and Communication Technology (ICT) has increased the number of possible business configurations a company can adopt. In other words, firms can increasingly work in partnerships, offer joint value propositions, build-up multi-channel and multi-owned distribution networks and profit from diversified and shared revenue streams. This, however, means that a company's business has more stakeholders, becomes more complex and is harder to understand and communicate.

The existing business model ontology [Osterwalder 04] which we want to render more precise consists of nine elements. Namely, value proposition, target customer, distribution channel, relationship, value configuration, capability, partnership, cost structure, and revenue model (cf. figure 1). A Value Proposition is an overall view of a company's bundle of products and services that are of value to the customer. The Target Customer is a segment of customers a company wants to offer value to. A Distribution Channel is a means of getting in touch with the customer. The Relationship describes the kind of link a company establishes between itself and the customer. The Value Configuration describes the arrangement of activities and resources that are necessary to create value for the customer. A Capability is the ability to execute a repeatable pattern of actions that is necessary in order to create value for the customer. A Partnership is a voluntarily initiated cooperative agreement between the enterprise and another company in order to create value for the customer. The Cost Structure is the representation in money of all the means employed in the business model. The Revenue Model describes the way a company makes money through a variety of revenue flows. The mentioned elements are further detailed and decomposed in the existing business model ontology and their relationships described.

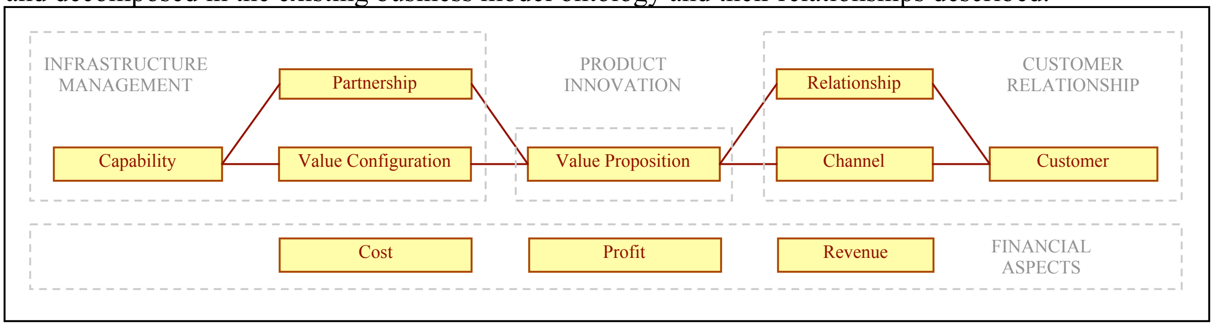

Figure 1: The business model ontology [Osterwalder 04]

Enhancing the business model ontology (= work in progress)

The proposition described above is indeed a model: It is independent of any enterprise and can describe the business model of any enterprise. Furthermore, it represents a synthesis of existing business model research. Therefore it complies with the first requirement of an ontology, being a generic description shared by a community of users. Yet, unfortunately no complete and definite consensus exists between the different communities of users. Some business model concepts include more or less so-called elements. The model outlined above represents a synthesis in the sense that it comprises all elements that have been mentioned by at least two different authors/researches in the field of business models.

The main goal of the business model framework is to provide users, such as managers, consultants process-, or IS/IT-designers with easy to understand, analyze, and compare descriptions of the business model of their enterprises. Although this goal has been reached and the proposed business model framework is far more precise in defining the relevant business model concepts than any other model it remains a first version. Even if we suggested an XML-based grammar for our ontology, for some business model applications, such as for computer-based business model design or requirements engineering some of the features of the proposed framework still need to be fixed more precisely. Indeed a formal definition of a model reduces the chances of misunderstanding and makes automatic translation into other languages possible. In particular implementation into a database management system or an ontology server can be automatically set up.

Our goals for an ontology of business models are not only to get an unambiguous model. We would like to complement our static description of business models with:

- integrity constraints, like "at least one value proposition of a company must be connected to a revenue stream that generates income from a specified target customer segment".
- derivation rules that define new concepts through formulae, like "a business model that only has free or low cost value propositions is a low-cost business model".
- a combination of both, like "eventually such a low cost business model must have an efficient infrastructure management".

Moreover we plan to offer users the possibility to query, analyze, and compare the descriptions of the business models of different enterprises that will be stored in an application.

Our reference business model has been described using an ad hoc meta model which is a kind of object-oriented data model. It supports the usual concepts of object class (called element), attribute, composition link, and inheritance link. It supports two other concepts: 1) a generic relationship link, similar to a binary relationship (without attribute) of the entity relationship model [Chen], and 2) a facility to define sub-parts of the schema.

This last concept is similar to the external-schema notion of the ANSI three levels schema architecture [Tsichritzis 78] This meta model does not support integrity constraints, rules, or instances. Therefore we need to move on to another meta model.

Which meta model for the ontology of business model? (= summary of one activity)

The information systems community progressively introduced ontologies at different stages: the Bunge-Wand-Web (BWW) ontology [Wand 96] is largely diffused for meta-modeling at the level of system analysis. Many ontologies have been proposed for covering enterprise modeling using different, more or less formal, syntaxes [Fox 98] [Geerts 97]. A very small number of ontologies has been proposed for addressing the business side [Ushold 98] [Gordijn 03] [Alter 04], most of them use ad hoc formalisms for describing their model.

Moreover there are quite a wide variety of meta models coming from the research communities in artificial intelligence, knowledge representation, and databases. The artificial intelligence community defined logic languages, especially description logics with their associated powerful inference techniques. These languages have been extensively used as formal theories on which several ontology languages have been designed, like OWL or RACER. The database community has defined conceptual data models with visual diagrams and associated design CASE tools. The structure of these data models range from binary models (e.g. DOGMA) to extended entity-relationship models (e.g. MADS). In between description logics and conceptual schemas lie the frame-based models that have been defined by the knowledge representation community. These models support some limited reasoning mechanisms. KAON and PROTEGE are examples of this category. Below, we shortly compare the two extreme kinds of models, description logics and conceptual models. A more detailed comparison may be found in [Cullot 03].

Data modeling: One could roughly say that conceptual models are better at designing primitive concepts because they can describe more complex structures, closer to the real world, and because they support appealing visual diagrams and design tools. Some of them also support a few derived constructs whose instances can be automatically inferred. But, contrarily to models based on a description logic, they do not support constructs that designers can define by a logical formula without knowing where they will fit in the generalization hierarchy or even knowing the generalization hierarchy. On the other hand, most description logics rely on simple binary structures, but they offer to the users all the power of their logic reasoners. They allow users to define precisely new constructs by a logical formula as complex as needed. Constraints may also be defined by logic formulae (of type inclusion or equivalence). The inference mechanisms automatically check the consistency of the new definitions and constraints, deduce where the new constructs are placed in the generalization hierarchy, and infer their instances.

Instances: Description logic systems naturally adhere to the open world assumption, which assumes that present data is just the explicitly known subset of the valid data, and more valid data may be inferred by sophisticated reasoning. On the other hand, databases follow the closed world assumption, stating that only information that is present in the database (or derivable by explicitly defined derivation rules) is valid. Consequently, they do not need sophisticated reasoners to infer additional information.

Constraints: Checking the consistency of the set of constraints and checking the consistency between the constraints and the schema are tasks that can be performed automatically by the reasoners available in description logics. On the other hand, databases have a normative approach and it is not possible to define a schema that does not obey the meta model constraints. Databases having no reasoning facilities, they cannot check the set of integrity constraints, and moreover usually they don't have an integrated language for defining integrity constraints. For example, "if a distribution channel delivers a value proposition and reaches a customer group, then this value proposition targets this customer group".

Instances querying: Databases and description logics offer complementary functionality for instance querying. Databases systems usually provide powerful assertional query languages complemented with efficient query optimization tools. Description logic systems support a set of simple functions for accessing instances and derived facts computed by their inference engines.

In conclusion, as only logic languages support the definition of integrity constraints and derivation rules that are fully integrated in the ontology description, a description logic system seems appropriate for defining a new version of business model. We plan to use PROTEGE as a front end editor for defining the structure of the business model, and then move to a description logic system, like OWL or RACER for the definition of the integrity constraints and derivation rules.

Conclusion (= a position statement)

Our ultimate goal in setting up a formal ontology of business models with a description logic system is to check on a real example what are the exact benefits of an ontology with respect to a database for the different kinds of users: the designers of the ontology Tbox (or database schema) and the end users that will query, compare, and analyze the descriptions of business models entered in the ontology Abox (or database). In this comparison, we want to find out if there exist – and which ones – characteristics of business models that require an ontology or a database approach. For instance, as descriptions of business models will always be incomplete, will the open world assumption of description logics be useful?

At the end, our research is based on the assumption that modeling the business model (value proposition, customers, partners, value systems, revenues, …) of companies, and not only their enterprise models (actors, roles, activities, resources, conversation, goals …) should contribute to improve their interoperability. This claim has still to be demonstrated. It is one of the main research questions we will address in this project.

References

[Alter 04] Alter, S., "Architecture of Sysperento, a theory based ontology of the IS field" (2004), *working paper, submitted for publication*.

[Chen 76] P.P. Chen, "The Entity-Relationship Model - Toward a Unified View of Data", *ACM Transactions on Database Systems*, Vol. 1, No. 1, March 1976, pp. 9-36

[Cullot 03] N. Cullot, C. Parent, S. Spaccapietra, C. Vangenot, "Ontologies: A contribution to the DL / DB debate", in *VLDB Workshop on Semantic Web and Databases*, Berlin, Germany, September 7-8, 2003

[Fox 98] Fox, M. S., Barbuceanu, M., Gruninger, M. and Lin, J., "An organization ontology for enterprise modeling," (1998) In *Simulating Organizations: Computational Models of Institutions and Groups*(Ed, Gasser, L.) AAAI/MIT Press, Menlo Park CA, pp. 131-152.

[Geerts 97] Geerts, G. L. and McCarthy, W. E. (1997) "Modeling Business Enterprises as Value-Added Process Hierarchies with Resource-Event-Agent Object Templates," In *Business Object Design and Implementation* (Ed, Patel, D.) Springer-Verlag, pp. 94-113.

[Gordijn 03] Gordijn, J. and Akkermans, J. M. "Value-based requirements engineering: exploring innovative e-commerce ideas," (2003) In *Requirements engineering*, Vol. 8, pp. 114-134.

[Osterwalder 04] Osterwalder, A. "The Business Model Ontology - a proposition in a design science approach" (2004) Ph.D. thesis, In *Institut d'Informatique et Organisation,* University of Lausanne, Ecole des Hautes Etudes Commerciales HEC, Lausanne, Switzerland, 173 pages.

[Tsichritzis 78] D. Tsichritzis, A. Klug, The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Dabatase Management Systems, *Information Systems* Vol.3, N°3, 1978, pp. 173-191

[Ushold 98] Uschold, M., King, M., Moralee, S. and Zorgios, Y. "The Enterprise Ontology," (1998) In *The Knowledge Engineering Review*, Vol. 13, pp. 31-89.

[Wand 96] Wand, Y. "Ontology as a Foundation for Meta-Modelling and Method Engineering," (1996) In *Information and Software Technology*, Vol. 38, pp. 281-287.