

Usage of Semantic Transformations in B2B Integration Solutions

Leonid Shumsky, Pavel Shapkin and Viacheslav Wolfengagen
*National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Cybernetics and Information Security Department, Moscow, Russia*

Keywords: B2B, Integration, Ontologies, Conceptual Modeling, Semantics.

Abstract: We propose a computation-oriented approach to ontology-based B2B integration. The domain ontology is used within the integration platform as a scheme for internal data representation. The main idea is to represent the ontology in a form that enhances the usage of static and dynamic type-checking features offered by the programming language compiler or runtime environment. For these purposes the ontology is represented as a set of types and transformation functions. We study the resulting integration platform architecture and compare expressive power and logical inference possibilities to other ontology representations.

1 INTRODUCTION

Currently every organization needs to establish the data exchange with its partners, counterpart agents and clients. The amount of messages could exceed hundreds of thousands and the count of integrated counterpart agents could be more than several hundreds. It is impossible to quickly organize a new or reliably support the existing communication under such circumstances without the usage of tools that accumulate knowledge about methods of interaction with specific systems as well as about process and data semantics in the corresponding domain.

The B2B integration specifics consists in the fact that the communication is to be carried out between the organizations that are neither organizationally connected, nor share common information systems. On the other hand, many organizations face the same integration tasks. These restrictions justify the requirement to use some reusable “external” description of processes and message structures (data blocks) that is standard for an industry or for a distinct task. Such standards are known as B2B integration standards. Nevertheless many tasks correspond to a set of different process and data description standards implemented by different applications (Cui et al., 2002; Jridi and Lapalme, 2013). The need to translate the data representation between different standards complicates the deployment of integration solutions. In such cases purely syntactic as well as semantic problems appear — it is needed to provide the full transition of semantics from the initial to the

target message. This problem can occur because with a single B2B message one can send more information than it explicitly contains: the message type, the delivery context etc. are also of great importance. Furthermore different standards use different restrictions on data types and different dictionaries for standard codes.

The main premise of the current work is that though the ontological approach to B2B integration is being sufficiently studied and implemented it still doesn't use all the possible advantages. We propose an approach that consists in the usage of a standard intermediate ontology not on the side of one of the systems being integrated but as a part of an intermediate system that provides services for data objects extraction and their transformation into the required format. The advantage of such approach is that applications that are connected as services and objects from the domain model which they provide are defined in a central system. Thus syntactic and semantic features typical for each concrete application are hidden from the user but nevertheless are fully implemented. Furthermore, such an approach enables a much faster deployment of standard B2B integration solutions because the connection between the applications and the ontology and the correlation between processes that take place in target systems are already defined.

The paper is organized as follows. Section 2 describes B2B standards along with their classification and specific features. Section 3 presents the essence of the proposed approach to the task of intermediate

ontology-based integration and the progress achieved by authors. Section 4 covers the specifics of organizing the B2B communication using domain ontologies. A simple example that uses the proposed formalization is given in section 5. The conclusion summarizes the key points and gives an overview of the related works.

2 B2B STANDARDS

The main aim of that section is determination and classification of B2B standards from the point of view of their content and scope. This classification is required by intermediate ontology construction algorithm. We should consider common integration standards and their specifics to determine which one will be used for ontology construction.

The set of B2B integration standards describes all integration steps, from technical characteristics of transported messages to descriptions of the target business process (Heravi et al., 2010). Table 1 shows some examples of standards and their relationships. This research considers levels 2 and 3, because the task of syntactic parsing for a well-known structure presents no difficulty and is not related to the semantics of the message. On the other hand, the task (problem) of parsing business process schemes, which are described by standards has its own specifics and is being solved in related works (see the conclusion section).

1	Syntax	XML, XML-RPC, Beep, SOAP
2	Messaging standards	ebMS, RosettaNet ImplementationFramework (RNIF)
3	Data structure standards	GISB, fPML, SWIFT, OBI, OTA, CXML
4	Business process structure	WS-CDL, XPD, ebBP, WE-BPEL
5	Business process templates	UBP, OAGIS Scenarios, RosettaNet PIPs
6	Business documents	UBL, OAGIS, RosettaNet

Despite of the similarity of tasks and their implementations' aspects, standards from different technological/instrumental stacks are not compatible. It means that even if some system supports integration for some business object with respect to one standard it can fail the integration for the same object but carried out in accordance with business process defined in another standard.

As Table 1 shows us, the most modern standards are based on open general-purpose standards developed by large, independent companies, like W3C,

OMG, OAGIS etc., which fact simplifies our task of parsing messages and extracting data from them. Important requirement for the message syntax is human readability. In case of program error, user should be able to change message and resend correct variant it manually.

We should discuss now each level of the model in details. Standards from the second level of the model prescribes specific features of messaging between systems. Standards of the third level determines data structures specific for some application, industry or business-object. Higher level standards are grouped in the same manner – they could be universal or industry specific. Nowadays, data structure standards are distributed as DTD or XSD documents or as WSDL service specifications to describe atomic acts of data transfer. But, application-specific standards may be described via low-level data structures and communication protocols, or even in terms of (remote) method calls.

Levels 4 to 6 determines schemas for typical or recommended integration business processes. A process is described by: its technological model (BPEL, XPD, etc.), suitable for execution with some specialized process engine; typical business process model (BPMN, UML, etc.); and high-level documents, which describe the approach to data exchange in general.

Summarizing, the intended integration schema for B2B integration includes following steps:

1. Business partners must come to an agreement on the integration of some business objects and choose any standard methodology, which has a description of each selected object;
2. Choose a required typical template for the integration process;
3. Partners must produce a technology model of the process, corresponding to the selected methodology, and provide its implementation ;
4. Partners must come to an agreement on the set of the standard messages, which will be involved in the process;
5. Messaging implementation must be provided.

Based on the list of standards and typical integration schemes, we can state that ontology-based approach may be applied to the fourth and the fifth steps. On the fourth step it will be used for the data semantic connection and errors investigation. On the fifth step it will be used for dynamic validation and incorrect messages filtering. The latter is particularly

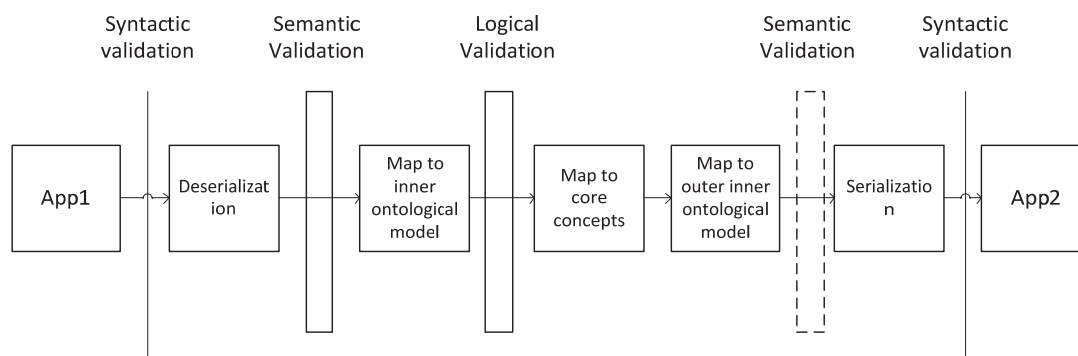


Figure 1: The message flow.

important, because syntactic structure of the messages is strictly defined and simple validation will not bring expected benefits.

In section 4 we will discuss the specific requirements to the ontology suitable for such a context: for semantic message validation inside integration processes and the main purpose of that ontology.

3 THE ESSENCE OF THE PROPOSED APPROACH

Approach, proposed in this paper, is based on the idea of creation of intermediate conceptual model for subject domain to provide independent data representation and transformation rules.

Concepts in the ontology are organized depending on their respective represented business object categories and (related to this object) processes. In general, such an intermediate conceptual model is a multilevel ontology. The top level contains descriptions for typically encountered (in various systems/domains) objects, grouped by their field of use. The second level corresponds to supported B2B methodologies, the third one – to specific data structures, within each methodology. The last, fourth, level reflects standards' features specific to particular applications.

Data transfer from one representation level to another is performed by a set of bidirectional transformations, which allows using a single implementation of the standard for both receiving and sending data. The main advantage of the proposed architecture is a reduced average operation's cost, due to reducing amount of the most expensive operations. The system of core concepts is rarely altered, as far as the task of connecting a new system is concerned. Adding a new standard means defining the third-layer ontology for its objects and the set of bidirectional transformations for that standard to core con-

cepts. Adding a new message type or a support for new application normally would not require doing extensive semantic modelling (so far as the application in question stick to the standard), merely introducing some syntactical extensions to the intermediate ontology.

The message flow in the proposed system is schematically shown in figure 1.

The message flow starts from the data structure of the fourth level received from the application and deserialized to the structure defined on the third level. These steps include syntactic validation by the scheme for that application and semantic validation for the stated standard. This step helps to eliminate messages that do not correspond to the standard semantically or syntactically. E.g. messages with inconsistent dates or undefined dictionary data fall in this category. The next step is the correlation of the data with the root concept which is the key advantage of the proposed approach. It includes standard-agnostic checks which accumulate expert judgments in a given domain. This step eliminates messages with deep inconsistencies which allow to assert that the data is erroneous or intentionally compromised. Such rules are set up for each copy of a system independently but the setup process rules from other installations or external sources can be used.

The reverse dataflow is symmetrical to the input flow. Root ontology data are serialized into a standard-specific and sent to the target application. On this step an additional standard compliance or syntax check for the outgoing message can take place.

Since the proposed approach is intended for a wide integration standard support concrete implementation of the systems being integrated does not affect the data transfer settings. It is only needed to provide the data access properties. All the data validation and transformation logics is hidden from the users but can be retrieved and altered if needed.

4 SEMANTIC APPROACH TO B2B INTEGRATION

Even if two organizations use the same standard there is no guarantee that the communication will be productive. The main source of the problems is that all the integration standards define only the structure of business processes and syntax structure of data and do not mention the semantics. It alters the goal of the ontology usage: usually the ontology defines document structure but in the case of B2B integration, when the document structure is already known, the ontology is used to implement complicated inner data validations.

4.1 Structure of the Domain Ontology

An ontology is a semantic model of the domain created as the result of *classification*. The process of classification consists in grouping of information objects (*individuals*) based on their similar features. Groups obtained in this process represent *concepts* which are abstract entities characterized by their so-called *intension* and *extension*. An intension is a rule used to determine whether an individual belongs to this concept. Individuals that satisfy this rule are called *instances* of the concept. An extension is a set of individuals, viz. instances of the concept. Concepts are organized in a hierarchical structure by the *subsumption* relation that forms a partial order on concepts.

The hierarchical structure of concepts is called *taxonomy*. An ontology is a taxonomy equipped with a system of declarative knowledge. In computer science, the ontology of the domain is an explicit specification of its conceptualization (Gruber, 1993). Knowledge that is added to a taxonomy can be seen as a set of rules that help to infer new knowledge about concepts from the existing knowledge. In order to construct provably correct inference procedures for ontologies we need a formal model of its constructs. The mainstream way to formalize ontologies today is to use logical formalisms. Such approach is used in the semantic web: web ontologies are formalized with description logics (Baader et al., 2007) that are fragments of first order predicate logics (FOPL).

From the point of view of business process integration, concepts of the domain ontology represent domain entities (e.g. Customers, Organizations, Opportunities, etc. considering the CRM domain). Moreover, for a domain entity concept, there may be a set of more specific concepts that correspond to representations of this entity in different applications

using different standards; each of these concepts should subsume the main entity concept. From this point of view we can prescribe different semantics to concepts regarding their position in the subsumption hierarchy. “Leaf” concepts correspond to specific representations of the domain entities in different systems and standards. Other concepts, which we will call *core concepts*, correspond to semantic representations of domain entities. Core concepts are used internally in the ESB to represent the data.

4.2 Transformation Semantics of the Domain Ontology

Our goal is to maximize the usage of the ontological information in the process of construction, setup and execution of the integrated business processes which are examples of computational objects of a special kind. Let’s consider how domain ontologies may be interpreted from the computer science and programming perspective. Ontology is a form of metadata, viz. “data about data”: it’s the additional data needed to interpret data objects. The basic way to represent metadata in programming languages is to use *types*. The advantage of type systems is that it is metadata that could be used by the compiler or the runtime environment to reason about programs, i.e. to statically or dynamically prove some properties of a program: absence of erroneous constructions, data security (Sabelfeld, 2003) etc.

From the programming perspective, concepts correspond to types. As for subsumption relations, they could be represented as transformations between different types: if the concept *A* subsumes the concept *B* then one can transform each instance of *A* to its “view” as instance of *B* and vice-versa, if there is a transformation from the objects of type *A* into objects of type *B* it means that one could assert that *A* subsumes *B* respecting this transformation. This approach to inheritance is known as *subtyping by coercion*. The equivalence of two concepts is understood as a special case of subsumption: *A* is equal to *B* iff *A* subsumes *B* and *B* subsumes *A*. Nevertheless there is one additional restriction on corresponding transformations: the compositions $tr_{AB} \circ tr_{BA}$ and $tr_{BA} \circ tr_{AB}$ should be identity functions on *A* and *B* correspondingly.

We’ve intentionally relied on transformations instead of inheritance relationship that could be used to model subsumption in object-oriented languages. Usage of transformations gives more possibilities as inheritance could be viewed as a special case in which the transformation function is just a trivial typecast to the base class.

Functional composition can be additionally used to extend the domain ontology by computing derived relations. Different transformations (e. g. $A \rightarrow B$ and $B \rightarrow C$) can be composed thus forming a chain of compositions that is a composite transformation (from A to C). From the logical point of view enabling the usage of composite transformation is equal to treatment of subsumption as a transitive relation.

The relationship known as the Curry–Howard isomorphism allows interpreting typing as proving theorems about programs, with various type systems corresponding to various logical systems. First order predicate logic (FOPL) is associated with sufficiently complicated type systems; nevertheless today more and more languages support these systems. In other words, the language of type systems is as expressive as the language of logical formalisms. It guarantees that there is a way to express ontological structures at the type-level of modern programming languages without any loss in the expressiveness and logical inference possibilities while making ontological information fully “understandable” and “usable” for the program language environment (i.e. compiler, interpreter, virtual machine etc.).

4.3 Usage of the Ontological Knowledge in the Integration Process Construction and Execution

The availability of the domain ontology enables to construct high-level descriptions of integrated processes

Usage of the Ontological Knowledge in the Integrated Process Construction and Execution

The availability of the domain ontology enables to construct high-level descriptions of integration processes which are bound not to concrete applications (e.g. Salesforce, AmoCRM, NetSuite, OpenERP, etc.) but rather to application classes (CRM, ERP, etc.). Data flows are expressed in terms of core concepts. The communication with concrete external systems is implemented according to the following rules:

- concept X instances can be sent to the system A if this system accepts the instances of X or more general concepts
- concept X instances can be obtained from the system A if this system provides the instances of X or more specific concepts

The rules mentioned could be used within the integration platform in the following ways:

- to validate the connections of the integrated process to external systems
- to generate tips on the set of available operations and object types while creating new processes

5 EXAMPLES

Let’s consider a simple integration scenario. Suppose we need to exchange sale orders between two point-of-sale applications PS_1 and PS_2 and two different ERP (ERP_1 and ERP_2) systems. There are different ways to formalize the corresponding ontology in terms of object types and transformation functions.

In the most “structured” case, if we are starting the development from scratch, we can take the top-down approach by creating a single data type T which accumulates order information which is common to all systems. Then, we have to define mappings which parse and generate representations of the information formatted for the used systems. If we need only to migrate orders from point-of sales to ERPs we need to define four mappings: two parsers from point-of-sales and two generators into ERP formats. For bidirectional migration eight mappings are needed.

Another possible scenario is when it is already known how to migrate data from PS_1 to ERP_1 using a type T_1 as internal data representation and from PS_2 to ERP_2 using T_2 . It is the case of the bottom-up approach when we first fully implement some migration functionality between concrete systems and then generalize it to connect all the systems.

In other words, suppose that we have four legacy mappings:

- $pt_1: PS_1 \rightarrow T_1$;
- $te_1: T_1 \rightarrow ERP_1$;
- $pt_2: PS_2 \rightarrow T_2$;
- $te_2: T_2 \rightarrow ERP_2$.

Using the proposed approach it is only needed to define two (or one *bidirectional*) transformations between the types t_1 and t_2 (named tr_{12} and tr_{21}) in order to construct a system which is semantically equivalent to the previous solution: the required missing migration functions are available as compositions:

- $te_2 \circ tr_{12} \circ pt_1$ migrates data from PS_1 to ERP_2 ;
- $te_1 \circ tr_{21} \circ pt_2$ migrates data from PS_2 to ERP_1 .

The equivalence of this solution and the previous one is proven by the fact that the ontology is semantically the same. The only difference is that we use a pair of equivalent concepts T_1 and T_2 instead of a single concept T . The equivalence of T_1 and T_2 is justified by the availability of the bidirectional transformation between them. Thus, using the proposed approach the domain ontology naturally arises from programming concepts: types and transformation functions. The ontology can be constructed top-down in the beginning of the project or it can be learned bottom-up from a set of types and transformations between them obtained from legacy systems. Further, this ontology is used to compute new ways of transforming the data and thus connecting different systems.

6 CONCLUSION

In comparison to similar solutions that use ontologies in the B2B integration domain (Haselwanter et al., 2006; Bussler, 2001) the proposed approach helps to abstract from the specific applications that take part in the integration process and to focus on the support of the standards used. In turn this not only makes the maintenance and the connection of new systems easier but also opens new perspectives in the field of data verification in terms of standards conformity, validation and consistency checks.

Related work. In order to have a richer B2B integration implementation from the beginning of the process to the technical deployment authors carry out research in the field of information process formalization in order to reveal connections with technological models and conceptual domain models (Wolfengagen et al., 2013; Shumsky et al., 2013; Shapkin and Demchenko, 2014). The combination of the integration process modeling according to B2B methodology standards and ontological data transformation approach described above enables to fully implement a B2B integration system. The deployment of such a system requires only to define the access points for the given applications in order to implement a complete, standard, recommended business process. In course of the execution such system could monitor and validate both the data (taking into account the history of requests) and the process structure.

REFERENCES

- Cui, Z., Jones, D., O'Brien, P., 2002. Semantic B2B Integration: Issues in Ontology-based Approaches. In *ACM SIGMOD Record*. Vol. 31, Issue 1. ACM New York, NY, USA, DOI: 10.1145/507338.507347.
- Haselwanter, T., Kotinurmi, P., Moran, M., Vitvar, T., Zaremba, M., 2006. WSMX: A Semantic Service Oriented Middleware for B2B Integration. In *Lecture Notes in Computer Science* Volume 4294, pp 477-483, Springer, DOI: 10.1007/11948148_43.
- Bussler, C., 2001. B2B Protocol Standards and their Role in Semantic B2B Integration Engines. In *Bulletin of the Technical Committee on Data Engineering*, 24(1), IEEE.
- Jridi, J.E., Lapalme, G., 2013. Adapting RosettaNet B2B Standard to Semantic Web Technologies. In: *15th International Conference on Enterprise Information Systems (ICEIS 2013)*. Pp. 443-450. DOI:10.5220/0004434404430450.
- Rahmanzadeh Heravi, B., Bell, D., Lycett, M., Green, S., Snelling, D., 2010. Towards Ontology Based E-Business Standards. In *Proceedings of the International Conference on e-Business*, pages 169-175, SciTePress, DOI: 10.5220/0002995901690175.
- Gruber, T.R., et al., 1993. A translation approach to portable ontology specifications. *Knowledge acquisition* 5.
- Baader, F., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F., 2007. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.
- Sabelfeld, A., Myers, A.C., 2003. Language-based information-flow security. *IEEE Journal on Selected Areas in Communications* 21, 5-19. doi:10.1109/JSAC.2002.806121
- Shapkin, P., Demchenko, A., 2014. Ontology-Based Type-System Integrated Tools to Infer Facts about Information Objects for .NET Platform. In *Autom. Doc. Math. Linguist.* 48, 52-59 DOI:10.3103/S0005105514020034.
- Wolfengagen, V., Roslovtsev, V., Shumsky, L., et. al., 2013. Applicative Approach to Information Processes Modeling - Towards a Constructive Information Theory // In *the proceedings of 15th International Conference on Enterprise Information Systems (ICEIS 2013)*. pp. 323-328.
- Shumsky, L., Roslovtsev, V., Kazantsev, N., et al., 2013. A synthetic approach to building a canonical model of subject areas in the integration bus // In *the proceedings of ISKO-Maghreb, 2013 3rd International Symposium (ISKO 2013)*: IEEE Xplore : Marrakech, Morocco - C. 1 - 7; DOI 10.1109/ISKO-Maghreb.2013.6728118