Canonical Data & Process Models for B2B Integration

Dumitru Roman
DERI Innsbruck
Technikerstraße 21a, 6020 Innsbruck, Austria
+ 43 512 507 6463
dumitru.roman@deri.org

ABSTRACT
In this document, we introduce the concepts of Canonical Data and Canonical Process Models as a way of tackling the problem of interoperability between widely adopted business standards (as far as business messages and respectively business processes are concerned). We describe the general problem that interoperability addresses in this context, and introduce a concrete example to see where such canonical models would be useful in practice. We then further analyse canonical data and process models as far as related work and potential solutions for building such models in a systematic way are concerned.

Categories and Subject Descriptors
D.2.12 [**Software Engineering**]: Interoperability. H2.1 [**Database Management**]: Logical Design. I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods.

General Terms
Design, Standardization, Languages

Keywords
Business standards, Interoperability, Canonical Data Models, Canonical Process Models

1. PROBLEM DEFINITION AND MOTIVATION
This paper tackles the general problem of data and process integration of already established, widely used business standards. The final aim is to provide a mechanism to ease the usage of different business standards in a homogenous way, thus providing higher value and faster integration for business processes, in general. For the sake of being concise, when we refer to business standards, we specifically refer to RosettaNet[1], UN/EDIFACT[2], and ebXML[3], however, the issues presented in this paper are applicable and can be generalized to any messages and protocols business standards. The following issues characterize the current business standards as far as messages and protocol specifications are concerned:

- some business messages already have (probably incomplete) XML Schema representations (e.g. the case of *some* RosettaNet messages), some have only DTD representations (e.g. all RosettaNet messages), some do not have any XML based representations (e.g. the case of EDIFACT) but only informal descriptions, and some do not commit to any message, but provide mechanism to include different message representations, e.g. XML based, or completely unstructured (e.g. the case of ebXML)
- some protocols are specified informally, in text documents (e.g. the case of all RosettaNet PIPs and EDIFACT communication protocols), some protocols are specified in some process languages (e.g. some RosettaNet PIPs are specified in ebBPSS), and some other standards only define the schemas that would allow different business protocols to be represented (e.g. UML activity diagrams in the case of ebXML).

What is the problem?

The *problem* can be generalized from the above mentioned issues:

- many different message formats that are semantically similar and
- many different protocols for achieving semantically similar tasks.

Such a diversity in message and protocol formats increases the integration costs of different applications and hinders adaptability and flexibility of business processes. Very often is the case that one company needs infrastructures that support different messages formats and protocols in order to collaborate with other companies; usually such infrastructures are expensive, and integrating them is even more expensive and time consuming. The human factor plays a key role in this integration process as he needs to understand sometimes ambiguous specifications of messages and process from different business standards and make them work together. Actually what the human does in this process is building a common understanding and models of similar messages and processes specified by different standards in different ways, and thus enabling interoperability through this common model. However, the problem is that, so far, such common models are represented implicitly, in humans' mind, designed in ad-hoc ways, with no clear methodology for building them, and thus hindering a faster and less expensive integration of different business standards.

[1] http://www/rosettanet.org
[2] http://www.unece.org/trade/untdid/
[3] http://www.ebXML.org

Proceedings of the VLDB2006 Ph.D. Workshop
Seoul, Rep of Korea, 2006

Why a canonical model is helpful for data and processes integration?

An *explicit* representation of such common models would open doors for a more organized way and more clear methodologies for building such common models. Having an explicit representation of these common models would allow integration of different business messages and protocols to take place in a more automated way, shortening the time and costs of integration. We refer to such explicit common models for messages and protocols specified in business standards as *canonical data* and respectively *canonical process* models.

It is the aim of this work to provide a general methodology to support the design of canonical data and process models from existing business standards. Once having such explicit common models, the key question here is how much in the design process of such common models can be *automated*, thus allowing for a much faster integration of existing business standards. This certainly implies the usage of some sort of formal techniques to provide support during the construction of these models; this is also part of our investigation.

One of the key issues is the representational language for the canonical models: what type of language is needed to enable flexibility needed to cope with the different representations of the messages and protocols of different business standards? The language should be general enough to allow for easy integration of different standards, as well as not committing to any specific formalism, but at the same allowing for an easy mechanism for attaching different formalisms/semantics to the representational language; the reason for initially not committing to any specific formalism is that the representational language for the canonical model should be flexible enough to be used with different formalisms (both static and dynamic aspects), and the user (i.e. the data/process integrator) should consider the semantics that suits his needs, depending on the reasoning tasks the s/he needs to perform (for example data or process transformation). One can argue that different static or dynamic languages have different expressivity (as well as corresponding complexity), and thus a semantics in a specific formalism for the canonical models might not always be possible – in this case we argue that that specific formalism is not suited for formally representing the respective messages and protocols. Moreover, the key point we are interested in here is not only the formal semantics of the canonical models, but the support that canonical models offer to the transformations of different messages and protocols (i.e. in which way such a language impact the mechanisms for semi-automatically translating the messages and protocols), and whether the canonical models can capture the expressivity expressed in the business standards. Because initially no formal semantics is meant to be attached to the representational language for the canonical models, we refer to this language as an *abstract* language. In this context, Figure 1 below highlights the place of canonical models in between the worlds of business standards and the formal languages for representing data and process languages.

The lowest level in the figure below presents the informal and semi-structured representations of standardized business messages and protocols. The middle level represents the common models that need to be built for or from the messages and protocols at the lowest level; this layer is meant to be expressive enough to capture the expressivity of all data and processes described in the business standards, as well as providing support for semi-automatic data and process mappings, by attaching the highest level, i.e. the semantic layer.

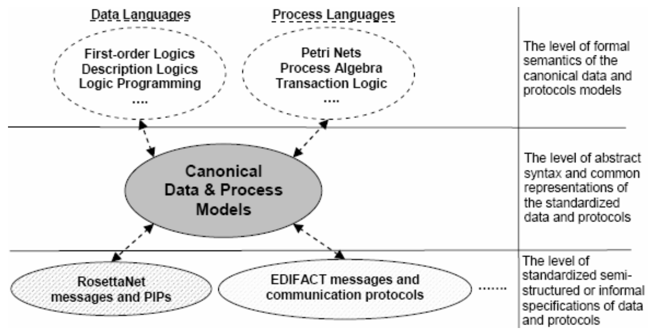

Figure 1. Canonical Data & Process Models – between formal languages and business standards.

The upper level in the picture represents different semantics that could be attached to the canonical models. Placing canonical models in between the informal world of business standards and the formal languages would enable the connection of the business data and processes to the formal languages in a flexible way. Once the canonical models are in place, and the relations between them and the different standardized business messages and protocols are established (i.e. the links between the lower and the middle layer in the figure above are defined), the semantics associated to the canonical models (i.e. the links between the upper and the middle layer in the figure above) could be used as a mechanism to support interoperability between the different standards.

An example

In order to get a better understanding in the role canonical models play in a typical business integration scenario, we give a short example in Figure 2.

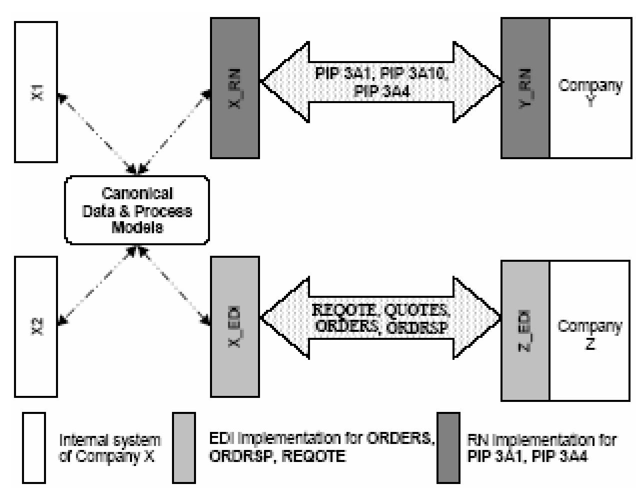

Figure 2. The place of Canonical Data & Process Models in a typical B2B example.

The figure above shows a company X requesting quotes and purchase orders from two other companies, Y and Z. Y and Z use RN messages and PIPs, and respectively EDI messages and communication protocols for providing quotes and order confirmations. Quotes and Purchase Orders RN and EDI messages have different structures and representations. The same is valid for the protocols used to request the quotes and the purchase orders; for example to request quotes, X_RN usually uses a sequential combination of PIP 3A1 (“Request Quote” PIP, which is further represented as a sequence of a request and response message) and PIP3A10 (“Notify of Quote Acknowledgment” PIP), whereas X_EDI uses as a communication protocol a simple sequential combination of REQOTE and QUOTES messages. Usually company X has processes that integrate requesting quotes and purchase orders from both Y and Z (e.g. a request for quotes is sent to different providers, and based on some internal decision, a quote is selected and then a purchase order is issued for that quote); in order to provide a flexible way to deal with the different RN and EDI messages, as well as with the different protocols (i.e. RN PIPs and EDI communication protocols), a common model for the quotes and purchase orders messages is needed, as well as a common representation of the RN PIPs and EDI communication protocols. Moreover, company X usually has internal systems (e.g. X1 and X2 in the figure) with their own data formats and protocols which need to be integrated with X’s RN and EDI system. The canonical data and process models are placed in between the systems that implement the different message and protocol specifications; building canonical data models imply designing a reusable mechanism for integrating business messages and protocols.

In this section we have described and motivated the general problem we tackle, and introduced the general concepts of canonical data and process models. The rest of this paper is organized as follows. Section 2 highlights relevant related work, and points out several deficiencies of related approaches. Section 3 provides some preliminary discussions that can be consider as a starting point for a potential solution for providing assisted support in building canonical data & process models. Section 4 concludes this document and indicates further steps.

2. RELATED WORKS

We are not knowledgeable of any attempt to tackle in a *systematic* and *unifying* manner (i.e. within a single framework) the problem of standardized business messages and protocols integration, and thus we will point out some works that tangentially touch and are relevant to the main issues we are interested in. The related work can be characterized in two types: industry driven attempts to align different business standards, and attempts done in academia, not directly related to business standards, but related to the more general problems of data and protocol integration.

On the industry driven attempts side, one relevant approach is OASIS Universal Business Language[4] (UBL). UBL defines a generic XML interchange format for business documents; it defines a royalty-free library of standard electronic XML business documents such as purchase orders and invoices. The main reason for developing UBL was that developing and maintaining multiple versions of common business documents like purchase orders and invoices is a major duplication of effort. However, it is unclear the relation between UBL documents and other business message standards such as RN or EDIFACT messages, as there have been no attempts to provide any transformations to such message standards. Without such transformations, UBL can be seen just another language for defining business documents/messages, and not a language that actually solves the problem of interoperability between different business standards. Moreover, UBL focuses only on the standardization of business data and doesn’t address the problem of protocols interoperability with other business protocol standards.

Another industry driven approach is Context Inspired Component Architecture (CICA)[5]. CICA is a business message architecture that provides an approach to business message development that facilitates a common reusable vocabulary across multiple industries. Although one of its design principle was alignment with other standards efforts (e.g. UBL), it unclear how CICA actually tackles the interoperability problem between different message standards as no explicit alignment is provided. As in the case of UBL, CICA doesn’t address the problem of business protocols interoperability.

The approach outlined in [1] applies to both the business messages and protocols and proposes an XML-based modernization and repackaging of business messages, and the use of a Web Services run-time stack to support message exchange and handling. WS-BPEL is proposed as a language for representing the different standardized business protocols. Although this approach tackles both message and protocol aspects of business standards integration, it doesn’t solve the problem: the generated XML messages as well as the generated WS-BPEL specifications for protocols are different for different business standards and no solution is proposed to the transformations of the messages and protocols (although they are represented in the same language).

Overall, the industry driven attempts to solve the integration of standardized business message and protocols are limited. All the proposed solutions are based on XML and different business messages are represented as different XML Schemas - the problem of integrating those schemas still remains. Moreover, the schemas very often differ significantly in the semantics of the concepts they represent, and besides a manual mapping (if at all possible), there is no mechanism to support the construction of common message schemas. On the protocol side, the situation is even worse, in the sense that there is not even an agreement on the language to represent the business protocols, not to mention about solving the problems of different representations of the semantically similar business protocols.

On the academia driven attempts side, the data and process integration techniques are relevant. Compared to the industry driven approaches, they address (semi-)automatic approaches to integration, which make them more relevant to our aim of building canonical models. In this context, the message/data approaches can be categorized in: *schema-based* approaches and *ontology-based* approaches. Although there has been a plethora of work in schema matching and mapping (e.g. [6]), it has been recently mentioned in [6] that ontology-based approaches are superior to schema-based approaches. Although several approaches have been proposed in this area (e.g. [8]), little has been done in the area of ontologizing the business standards (e.g. [4]), and thus it is unclear how one could actually use the

---

[4]http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

[5] http://www.disa.org/x12org/MEETINGS/x12trimt/cica.cfm

ontology-based approaches for mapping between different business standards. The question whether different ontology languages are expressive enough to represent different business standards, as well as providing assisted support for building canonical data models (i.e. lifting semi-structured message representations to ontological elements, and providing mapping support between the ontological elements) is still open. Nevertheless, it has been recently shown in [9] that ontology-based approaches provide better flexibility, and a more automatic way for providing mappings between ontologies, thus making ontology-based approaches to data integration worth looking at when building canonical data models from business standards. On the protocol side, the relevant academia-driven approaches can be categorized in: approaches that directly deal with business standards, and approaches that tackle, in general, protocol/process transformations. The work in [9] presents an approach to transform RN PIPs to Abstract WS-BPEL processes; it assumes WS-BPEL as a language for representing standardized business processes, however it is unclear if Abstract WS-BPEL is expressive enough to represent also other standardized business protocols. Even if Abstract WS-BPEL would be expressive enough, the problem of mapping and solving mismatches between WS-BPEL specifications is still an open issue. In [11], a communicative acts-based approach, combined with an event-guard-action model is taken to representing business protocols. It assumes that the requester and provider share the same communicative acts. But not transformation mechanism is presented to cope with heterogeneous abstract protocols that govern the sequencing of the occurrence of the communicative acts. Other process related works that are relevant in our context tackle, in general, the issues of protocol/process transformations. [3] presents some transformation rules between sequences of messages, however the set of transformations is rather limited as it only deals with sequence of messages and doesn't take into account more complex interactions. On the other hand, [12] and [13] deal with more complex transformations between more complex workflows, however they are limited to structured workflows and do not take into account temporal constraints – feature very often needed in practice.

What are the challenges?

In a nutshell, no previous work has tackled in a systematic way the integration of both messages and processes aspects of business standards. On the messages/data side, ontology base approaches look promising for their potential support for building assisted canonical data models for different standardized business messages. However, little has been done in ontologizing the standardized business messages, thus leaving open door for further research in this area. The key *challenge* here is to find a trade-off between an ontology language expressive enough to represent business messages on one side (and to provide assisted support for ontologizing business messages), and its assisted support for data transformation on the other side. On the protocol side, workflow transformations techniques seem promising to assist development of canonical process models for different standardized business protocols. However, such transformation techniques have been applied only to workflow languages limited in expressivity (e.g. which do not include support for temporal constraints), thus leaving open door for further investigations in this area. The key *challenge* here is to find a trade-off between a workflow language expressive enough to represent the standardized protocols on one side (and to provide assisted support for support for generating workflows from standardized business protocols), and its assisted support for protocol transformation on the other side. Moreover, integrating canonical data models and canonical process models in a single, unifying framework, is even a more challenging problem, that is the final aim of the work presented here.

3. THE BASIS FOR A POTENTIAL SOLUTION

In order to tackle the above mentioned problems, an approach based on generalization is taken. That is, in a first stage we are interested in identifying/building a conceptual model that would give us enough flexibility to express the elements of different standardized messages and protocols. For this, we take as a starting point the conceptual model provided by the Web Service Modelling Ontology (WSMO) [1]; WSMO is an ontology-based approach to representing data and Web services.

In a nutshell, as a data representation meta-model, WSMO essentially allows for defining concepts, relations, instances, and axioms. The model is general enough to capture static aspects of data/messages. One can easily argue that other meta-models for ontologies would be fitted for this, however what makes the WSMO conceptual model interesting for our problem is the fact that various formalizations exist for it in terms of the Web Service Modelling Language WSML[6] (which is based on different logical formalisms, namely, Description Logics, First-Order Logic and Logic Programming), and thus making (parts of) WSMO compliant ontologies easily mappable to other ontology languages supported by single formalizations. As it was shown in Figure 1, we are looking for a conceptual model that can capture the structure and constraints of the business messages, as well as being easily grounded in different formalization. WSMO/WSML can be seen as a promising framework for tacking this. Moreover, there already exists some work in ontologizing EDI in WSMO/L, which can be considered as a starting point for ontologizing other standardized messages as well.

On the process/protocol side, WSMO provides a general state-based approach for modelling interactions and processes[7]. However, such an approach can be seen as too low level, and not very intuitive for modelling high level business protocols/processes, where a more explicit representation of the control flow between tasks is needed. To tackle the issue of the conceptual model for the protocol specifications, we consider and adopt the work in [15], where, as a result of systematically evaluating the features of existing workflow systems and common recurring business requirements, a set of control flow patterns was compiled. Later on, these patterns were materialized in the YAWL language [16] – a language which offers direct and intuitive support for modelling all control flow patterns. Although control flow patterns represent core elements that are part of business protocols/processes and need to be reflected in our conceptual model, they come with one obvious limitation [14]: they can not be used to specify global dependencies/constraints between workflow tasks. In order for us to be general enough to

[6] http://www.wsmo.org/wsml

[7] http://www.wsmo.org/TR/d14/v0.3/

ensure that we can capture all the business protocols, temporal constrains also need to be reflected in our conceptual model.

In a second stage we are interested to look at the various formalisms to support the conceptual model that is meant to be provided in the previous stage. However this stage should not be regarded as an effort to provide a formalism for the conceptual model, but rather as a mechanism needed to support the third stage – the transformation of data/messages and protocols. Since the static aspects of the planned conceptual model are more or less solved (i.e. the ontology part of WSMO, and its formal semantics in WSML), our effort will concentrate on the dynamic aspects of the model. For this, we envision the usage of Concurrent Transaction Logic [17] as the underlying framework for formally representing the workflow patterns as well as temporal constraints that will need to be reflected in the model.

In a third stage, and the most important to us, the transformation techniques, based on the conceptual model and the formal framework resulted from the previous stages, will be developed. On the data aspect, [9] already provides an approach to assisted mappings between ontologies specified in WSMO/L, and our further investigation will go towards the techniques generating WSMO/L ontologies from business messages, rather than on the actual mapping techniques. On the protocol/process aspect, it is planned to investigate the applicability of the protocol transformation techniques in [12] and [13] to the model that this work should produce in the previous steps.

To summarize, our proposal is based on taking WSMO as a conceptual model, extending and integrating it with best of bread in workflow languages – that is, integrating workflow patterns into the model, as well as temporal constraints. Such an expressive model and its associated formal model would allow us to capture and represent any standardized business messages and protocols. Moreover, the formal foundations of such a model would allow us to apply and further extend existing approaches in data and process transformation (e.g. [12], [13]), thus providing a mechanism for building Canonical Data & Process Models for different business standards.

4. CONCLUSIONS AND OUTLOOK

In this document we have introduced the concepts of Canonical Data & Process Models and highlighted the key role they can play in achieving the interoperability of different business standards. In the context that up to 30% of future IT budgets will be spent on integration and interoperability issues, we believe that the problem is important enough to be tackled in a systematic way. Actually, no previous work tackled the interoperability of different standardized business messages and protocols in a systematic and unifying way (i.e. within a single framework). We have sketched the basis of a potential solution which incorporates a general, yet meaningful conceptual model (that borrows elements from ontologies, as well as from workflows) with formal languages on top of which assisted model transformations can be more easily applied.

5. ACKNOWLEDGMENTS

I would like to thank Christoph Bussler for his valuable feedback into this document. The work is partly funded by the European Commission under the projects ASG, SWING, and SUPER.

6. REFERENCES

[1] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel. Web Service Modeling Ontology. *Applied Ontology*, 1(1): 77-106, 2005.

[2] P. Bunter *et al*. An approach to moving industry business messaging standards to web services. Online at http://www-128.ibm.com/developerworks/webservices/library/ws-move2ws.html.

[3] E. Cimpian and A. Mocan. *WSMX Process Mediation Based on Choreographies*. 1st Intl Workshop on Web Service Choreography and Orchestration for Business Process Management, 2005.

[4] D. Foxvog and C. Bussler. *Ontologizing EDI: First Steps and Experiences*. Intl Workshop on Data Engineering Issues in E-Commerce, 2005.

[5] M. Nagarajan, K. Verma, .A. Sheth, J. Miller, J. Lathem. *Semantic Interoperability of Web Services – Challenges and Experiences*, LSDIS Lab Technical Report, 2006.

[6] E. Rahm and P. Bernstein. A Survey of Approaches to Automatic Schema Matching. *VLDB Journal*, 2001.

[7] S. Seyhmus. *Ontology Mapping Survey*. Available online at http://www.srdc.metu.edu.tr/webpage/seminars/Ontology/Ontology%20Mapping%20Survey.ppt

[8] Y. Kalfoglou and M. Schorlemmer. Ontology Mapping: the State of the Art. *The Knowledge Engineering Review* 18(1): 1-31, 2003.

[9] A. Mocan and E. Cimpian. *Mapping Creation Using a View Based Approach*. 1st Intl Workshop on Mediation in Semantic Web Services, 2005.

[10] R. Khalaf. *From RosettaNet PIPs to BPEL Processes: A Three Level Approach for Business Protocols*. 3rd Intl Conference on Business Process Management, 2005.

[11] K. Stuart. S. A. Battle, J. E. Cuadrado,. *Protocol Mediation for Adaptation in Semantic Web Services*. Technical Report HPL-2005-78, 2005.

[12] W. Sadiq and M. E. Orlowska. *On Business Process Model Transformations*. 19th International Conference on ER, 2000.

[13] J. Eder, W. Gruber, and H. Pichler. *Transforming Workflow Graphs*. 1st Intl Conference on Interoperability of Enterprise Software and Applications, 2005.

[14] H. Davulcu, M. Kifer, C. Ramakrishnan, and I. Ramakrishnan. Logic based modeling and analysis of workflows. *ACM Symposium on Principles of Database Systems*, 1998.

[15] W. M. P. van der Aalst, A. H. M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1): 5-51, 2003.

[16] W. M. P. van der Aalst and A. H. M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4): 245-275, 2005.

[17] A. J. Bonner and M. Kifer. *Concurrency and Communication in Transaction Logic*. Joint Intl Conference and Symposium on Logic Programming. MIT Press, 1996.