eChallenges e-2009 Conference Proceedings
Paul Cunningham and Miriam Cunningham (Eds)
IIMC International Information Management Corporation, 2009
ISBN: 978-1-905824-13-7

A Framework for Managing the Complexity of Business Document Integration

Christian EIS[1], Christian HUEMER[2], Philipp LIEGL[2],
Christian PICHLER[1], Michael STROMMER[1]
[1]*Research Studios Austria, Thurngasse 8/20, Vienna, 1080, Austria*
Email: {firstname.lastname}@researchstudio.at
[2]*Vienna University of Technology, Favoritenstraße 9-11, Vienna, 1040, Austria*
Email: {lastname}@big.tuwien.ac.at

Abstract: The electronic exchange of business documents is an often communicated desire by enterprises and the public sector. Electronic business is currently mainly hindered by unresolved interoperability issues and a lack of tool support. Additionally, computer aided document exchange is cost intensive at the beginning and thus discourages small and medium-sized companies to adapt to new technologies. The contribution of this paper is to present suitable tool support and methods to manage the complexity of business document integration. Our modeling tool is based on UN/CEFACT's (United Nations Centre for Trade Facilitation and Electronic Business) Core Components Technical Specification allowing for platform independent, conceptual business document models. On these models further integration operations can be performed.

1. Introduction

Electronic Data Interchange (EDI) is still one of the key challenges for today's enterprises. Although the idea of exchanging business documents came up quite early, lots of interoperability issues have not been resolved yet. Moreover, many companies, despite being capable of implementing EDI standards, are reluctant to invest their money in EDI solutions. The reasons for this are manifold.

Firstly, there is little to no tool support for handling business document models on a conceptual level. Many business document standards make use of the XML syntax - common tools for visualizing XML Schemas are considered best practice. When working on large models, navigation and manipulation soon become time-consuming tasks. Without proper manageability a sustainable development and usage of business documents is unlikely to be achieved.

Secondly, for the time being there exists a vast amount of XML business document standards [1]. Examples of such standards or de-facto standards are UBL [2], HL7 [3], SWIFT [4], and HR-XML [5]. They differ in spreading, which is in most cases unknown, size, and often in business domain. Thus, it is hardly possible for small and medium-sized businesses to choose the right standard, also not knowing which one will prevail in the future and which one will not.

Thirdly, the integration of a new business document standard into an existing information system is a tedious and error prone task. Integration problems may arise from various sources [6]. Definitely most complicated are those arising from semantic conflicts, when two data models are being aligned.

Fourthly, current business document standards follow a top-down approach in that they try to cover all concepts ever being used in a certain domain. As a consequence the models become too comprehensive and too hard to manage.

2. Objectives

Our goal is to provide methods and solutions to the user for the above mentioned drawbacks and shortcomings in business document engineering. We aim at developing *concepts for document model mapping and transformation* in order to support business document engineers in their everyday work. More specifically, we want to provide a set of tools, which ease the integration of electronic business documents in existing or new information systems. Upcoming technological paradigms such as service oriented architectures more than ever demand for a common methodology in order to define the underlying business documents and their data model. Current efforts and methods are often either inappropriate, too time-consuming or not sufficient in meeting the specific requirements of service oriented architectures. However, only efficient and easy to use solutions will enable a widespread use of services and their interactions in a service oriented architecture. Moreover, successive cuts in implementation cost may encourage companies to join EDI and thus leverage long term profits from this technology in general.

3. UN/CEFACT's Core Components at a Glance

Our approach is based on the latest Core Components Technical Specification (CCTS) [7] maintained by UN/CEFACT (United Nations Center for Trade Facilitation and Electronic Business). UN/CEFACT is a non-profit organization under the umbrella of the United Nations, developing standards for international trade facilitation. Within CCTS so called core components are the central building blocks. Thereby core components represent reusable parts with a common semantic meaning for assembling business documents. A core component does not have a specific business context, thus it can be used in any given scenario. In order to contextualize a core component to the specific needs of a given business domain the concept of business information entities is being used. Business information entities are derived from core components by restriction and are used in a specific business domain. Thus, core components represent the common semantic basis for all business information entities.

Having these concepts at hand it is possible to model business documents first in a conceptual, implementation and platform-independent way using core components. The created core component libraries can then be used to derive different libraries of business information entities for certain business scenarios.

A predefined global library of core components is currently developed and maintained by UN/CEFACT and has become known as Core Component Library (CCL) [8]. However, core components in the CCL are defined using spreadsheets and thus integration into modeling tools is difficult. In order to apply the core component concepts as a means of modeling business documents we have created a UML profile that allows using core components with standard UML modeling tools. This standard has been developed at the Research Studios Austria, Studio Inter-Organisational Systems[i] and has consequently been submitted to UN/CEFACT for standardization. Today, the standard has become known as the UML Profile for Core Components (UPCC) [9]. Because of limitations in the UML profile mechanism and the UML per se, we have been implementing additional concepts that are vital in the domain of document engineering. For a detailed description of UPCC and its additional semantics we refer the interested reader to [10].

4. Implementation

With CCTS and the UML Profile for Core Components (UPCC) as a foundation we are building a tool for document engineering called VIENNA Add-In (Visualizing Inter ENterprise Network Architectures) [11].

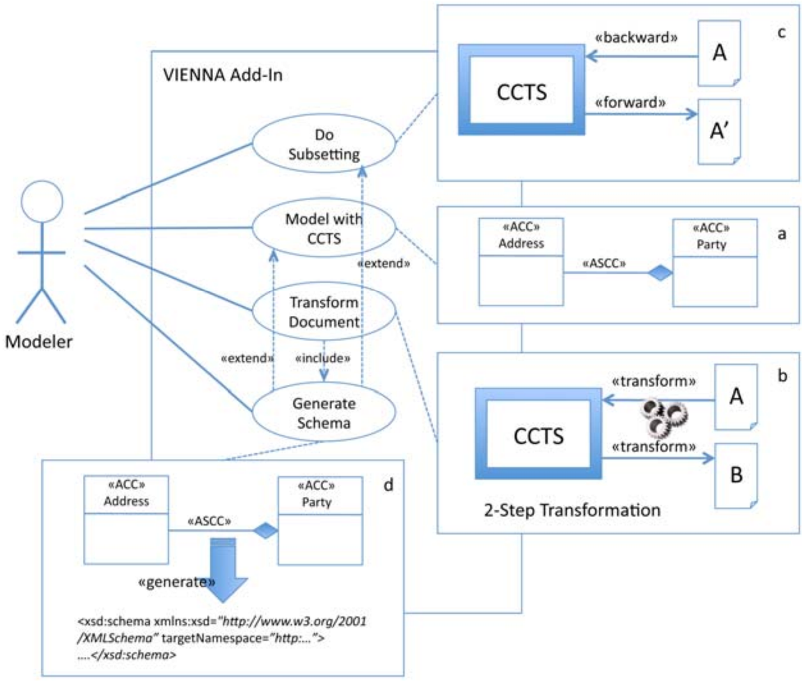

Figure 1: Overview VIENNA Add-In Use Cases

The VIENNA Add-In is an open-source plug-in written for the UML modeling tool Enterprise Architect. Enterprise Architect provides the main modeling features for UML 2.1 we rely on. Based on two main pillars, the VIENNA Add-In provides the business modeler with all necessary modeling, validation and generation functionalities. The first pillar of the Add-In concentrates on the exact exchange order of business documents in an inter-organizational business scenario i.e. an inter-organizational business process model. The second pillar deals with all necessary aspects of the business documents themselves. In this paper we will only concentrate on the business document modeling capabilities of the VIENNA Add-In.

4.1 Typical Use Cases in Document Engineering

The four main use cases identified during our research that shall be addressed by the VIENNA Add-In are depicted in Figure 1 and are described in the following paragraphs.

Model Documents. In the field of business document integration it is essential to have a proper modeling environment at hand. Modeling is the very basis of a successful integration task. One may either build a whole document model from scratch with intuitive and meaningful tooling or build upon existing models and just modify or expand them. As we allow the user to model documents by means of CCTS models, the modeler is offered a concrete graphical syntax, which is easy to handle and understand (c.f. Figure 1a). In contrast to a textual syntax, relationships between document parts, i.e., individual core components, are visualized to the user immediately. In a textual syntax concrete relationships between elements are often hard to find and not apparent at first sight. For example, the source code of an XML schema barely reveals the opposite of a referencing element and neither do tree representations of such schemas. The UML and its profile mechanism overcome some of the difficulties in visualizing models. This is however only true for small models, say 20-30 core components with connections among each other. For larger models it is often hard to align elements and connections in such a way, that connections do not overlap. In UML we have the possibility to reduce model growth by using packages for grouping purposes and to reduce the complexity of each sub-model.

Transform Documents. After having successfully modeled documents tailored to one's own needs it may be necessary to adopt these documents to some external requirements. For example a business partner uses some entirely different document model, but it should still be possible to do business. This is where document model transformations are put into play. Transformations in our VIENNA Add-In will be realized as 2-Step Transformations (c.f. Figure 1b). The precise semantics of a mapping and model operators used in this paper are defined in [12].

First Step. In order to reduce the necessary amount of transformations needed for a certain set of models to be transformable into each other, we introduce the CCTS conceptual model as a pivot model M_P. This transformation task from a model M_A to M_P is preceded by the definition of a comprehensive mapping model $map_{AP} = M_A \leftrightarrow M_P$, which relates objects in both models. This mapping model takes care of the proper alignment of the concepts used in models M_A and M_P. The mapping task has to be done manually by a domain expert. Automatic matching tools are considered optional to produce an initial mapping model. Mapping results of such tools may however be of inadequate quality if structural differences are too extensive [13]. Furthermore, matching is currently not in our research focus as we concentrate on design and integration methods from a more global perspective. As a visual mapping tool we make use of Map Force by Altova[ii] that offers a convenient user interface with various mapping functions. The intermediate transformation is then given by $M_{P'} = t(M_{A,}\, map_{AP})$ where $M_{P'}$ is a subset of M_P.

Second Step. To complete the transformation from a model M_A to a model M_B, model M_A expressed as CCTS model is transformed into model M_B. Again, it is assumed in this second step, that a mapping between model M_B and M_P, i.e., map_{BP}, describing the transformation rules, has been created, manually or semi-automatically. This second partial transformation is given by $M_B = t(M_{P',}\, map_{BP})$ leading to a complete transformation $M_B = t(M_A\, ,map_{AB})$ where map_{AB} can be computed from the composition of map_{AP} and map_{BP}. For a discussion of the compose operator on mappings see Bernstein [12].

Create Subsets of Documents Another very common use case in document engineering arises when two or more participating parties want to define a subset of a prevailing business document. Many business document standards follow a so called top-down approach and try to include as many business concepts as possible in order to handle any possible situation or business. This paradigm leads to overloaded and complex standards. UBL [2] is a perfect example for such an overloaded business document standard. Therefore, we are going to allow for a round-trip engineering process, producing a valid schema from the subset of a larger document model (c.f. Figure 1c). In the first phase of this round-trip an existing model M_A is backward engineered to correspond with the conceptual modeling layer of CCTS. Thus, the transformation $M_{P'} = t(M_{A,}\, map_{AP})$ is applied on M_A. Upon this conceptual layer restrictions to the model may be applied, leading from $M_{P'}$ to $M_{P''}$. Extensions are not permitted to the user, otherwise compliance with the original model is no longer possible and instances may be dropped during the exchange of business information in a business process. The subsetting process is finished through forward engineering the newly created model $M_{P''}$ by executing the transformation $M_{A'} = t(M_{P'',}\, map_{AP})$, where $M_{A'}$ is a valid subset of M_A.

Generate Schemas from document models. All three use cases described above may include the generation of corresponding XML Schema documents (c.f. Figure 1d). The task of schema generation may again be divided into three subtasks:

1. *Serialize CCTS model.* This use case may most likely occur after modeling an arbitrary business document from scratch. The serialization of the XML Schema is driven by the general Naming and Design Rules provided by UN/CEFACT [14]. By doing so we ensure that CCTS models conforming to these rules are interchangeable across platforms and applications.

2. *Serialize some model.* This use case is triggered during the transformation of say document model M_A into document model M_B. The serialization of the XML Schema of model M_B must adhere to the custom naming and design rules of the original document model on which the mapping from M_B to CCTS M_P was defined. The generated schema will in most cases only cover a subset of the original model M_B, otherwise the transformation becomes obsolete.
3. *Serialize subset of some model.* After a particular subsetting of model M_A to model $M_{A'}$ the generation of the schema follows specific naming and design rules as explained in the previous subtask. Furthermore, it has to be ensured that model $M_{A'}$ is strictly a subset of M_A and does not contain any extensions. Otherwise instances $o_{A'}$ conforming to $M_{A'}$ will no longer be conforming to M_A leading to further interoperability problems.

5. Tool Demonstration

In its current implementation, the VIENNA Add-In supports the following features:

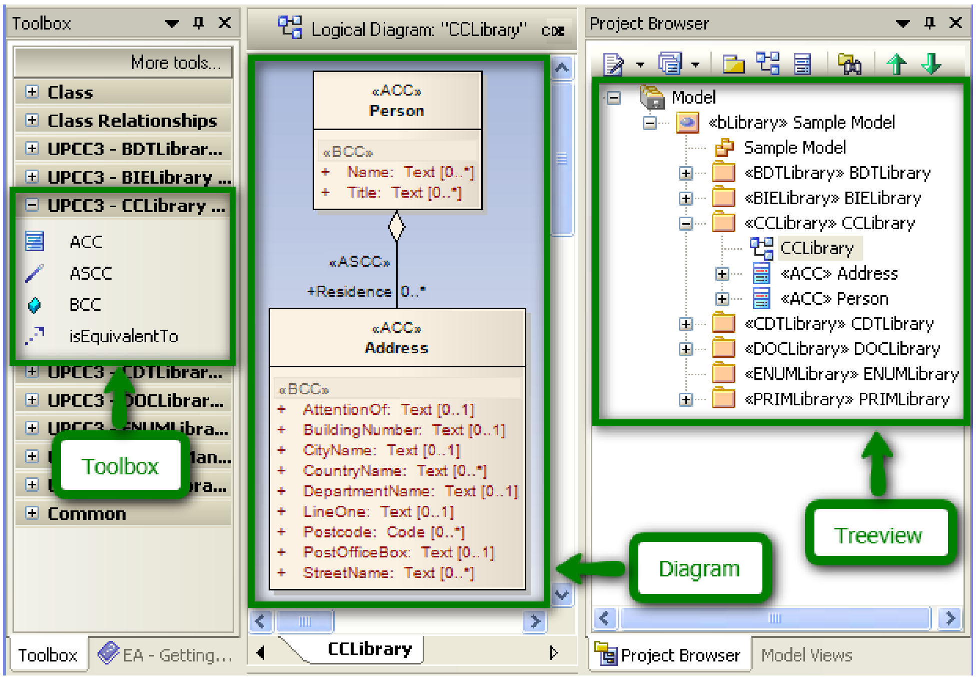

Figure 2: Screenshot of the Basic Modeling Interface of the VIENNA Add-In

Document Modeling. In order to provide support for document modeling and subsetting use cases as described above, we have integrated the UML Profile for Core Components (UPCC) in Enterprise Architect (EA) as a custom toolbox profile. This allows users to compose UPCC-compliant document models via drag-and-drop. Figure 2 shows the EA user interface components related to the basic modeling task. The core part is a class diagram for UPCC artifacts as shown in the center of the Figure. We distinguish between three different types of core components. An aggregate core component (ACC) represents a complex object that contains basic properties – so called basic core components (BCC). Association core components (ASCC) are used to describe complex properties i.e. associations between two aggregate core components. Likewise we distinguish between three different business information entities. Aggregate business information entities (ABIE) are used to represent complex properties. They consist of basic business information entities (BBIE) and association business information entities (ASBIE).

The core component example in the center of Figure 2 contains two aggregate core components (ACC), **`Person`** and **`Address`**, along with their respective basic core components (BCC) and association core components (ASCC). A new ACC (**`Party`**) can be added to the library by dragging the ACC item from the toolbox on the left to the diagram which creates a class with stereotype ACC. The user can then create BCCs by dragging BCC items from the toolbox to the newly created class. Finally, an association core component between two components (e.g. **`Party`** and **`Person`**) can be created by selecting the ASCC item in the toolbox and clicking and dragging from one class to the other. BCCs and ASCCs can then be further refined e.g. by specifying cardinality or role names for ASCCs. Additional CCTS metadata can be specified as tagged values defined by the UPCC toolbox items.

The tree view on the right hand side of Figure 2 shows the model structure, including available business libraries (e.g. business information entity libraries), as specified by UPCC. Each library package contains a class diagram which is used for modeling the structure of the components within the library in a similar way as described above for ACCs. In addition to this basic modeling support, we have implemented a number of wizards to support the user in more complex modeling tasks. At the moment these features include a wizard for creating and editing ABIEs, as well as a wizard for creating BDTs. An ABIE is based on an ACC, so the wizard allows the user to select the appropriate ACC from the model. In a next step the user selects all or a subset of the ACC's BCCs to be used as the basis for the ABIE's BBIEs. The user may then duplicate and/or rename selected BBIEs. Finally, the user must specify each BBIE's type by either selecting an available BDT or creating a new one. New BDTs are automatically created by the wizard, based on the corresponding CDTs by copying all of the CDTs' attributes, i.e., content and supplementary components. As for ASBIEs, the user can select and create them in a similar fashion on the Associations tab, based on the ACC's ASCCs. Also, the user may specify some general information, including a qualifier prefix, the name of the ABIE and the libraries to store the generated ABIE and optionally generated BDTs.

Validation In addition to the modeling support, we have also implemented validators to determine, whether a model satisfies the constraints defined by the UPCC. In the current implementation, the validation must be triggered by the user. However, since the validation of large models is a time-consuming process, we are experimenting with on-the-fly validation, similar to the code inspection features of modern software development environments. This feature would provide the user with instant feedback, whenever the model is changed and could also provide quick fix functionality, proposing (semi-automatic) steps to correct an error. For example, if the user adds a class to a CC library, without specifying the correct stereotype (ACC), the on-the-fly validation would notify the user and provide a quick fix to add the appropriate stereotype.

Schema Generation. Finally, in support of the schema generation use case, as well as the document subsetting use case, the Add-In provides functionality to serialize CCTS models as XML Schema documents, according to the UN/CEFACT Naming and Design Rules. Generation of other XML Schema documents, adhering to some arbitrary naming and design rules, is currently being worked on, as well as the use of Map Force mapping definitions for document transformations and round-trip engineering. Additionally, a case study will be conducted, which addresses these use cases both from a conceptual and a practical point of view. First results of this case study will be available at the beginning of next year.

6. Related Work

The work presented in this paper focuses on conceptual modeling of business documents, while utilizing UN/CEFACT's core components as a conceptual model. Another business document standard based on core components is the Universal Business Language (UBL) [2]. UBL defines a similar mapping from core components to real XML constructs compared to the Naming and Design Rules specified by UN/CEFACT.

Among the work in this area, [15] addresses the problem of structural as well as semantic heterogeneity between messages exchanged in the Web services domain. The idea of the approach is to resolve message heterogeneities by mapping message elements to conceptual models. However, the mappings are defined at the schema level, i.e. the logical level, between Web Service Description Language (WSDL) [16] and Web Ontology Language (OWL) [17] schemas while utilizing Web Service Semantics (WSDL-S) [18].

A similar approach is presented in [19]. The approach introduces a semantically enriched approach for dynamic data mediation in EAI scenarios based on ontologies, Semantic Web, and Semantic Web Services Technologies. Dynamic data mediation is achieved through transformations of the Web service inputs and outputs to a common reference model named Enterprise Interoperability Ontology (ENIO) [20]. The EAI ontology is an upper-ontology covering generic concepts with domain-specific extensions called facets. ENIO has a three-faceted structure including a data facet, which is based on UN/CEFACT's Core Components Technical Specification (CCTS).

7. Conclusions and Future Work

Earlier in this paper we described the major problems in the field of business document engineering based on concrete examples. To cope with these problems we propose a set of methods and tools which enables business document engineering in an efficient and convenient way. Therefore, we first created a UML profile named UPCC for UN/CEFACT's core components. The profile allows core components to be used with any standard UML modeling tool. Furthermore, we developed a tool supporting business document engineering named the VIENNA Add-In [11] which represents an extension for the UML modeling tool Enterprise Architect. The UML profile together with the Enterprise Architect allows to conveniently model core component compliant business document models on a conceptual level. Moreover, model transformations may be used to deal with the vast amount of XML business document standards as well as to resolve integration problems such as syntactical or semantical incompatibilities. Model transformations are currently only feasible on the conceptual level and a specific implementation is currently being worked on. In addition a generator has been implemented, allowing to generate XML schema artifacts based on business documents modeled on the conceptual level.

Currently, business document standards themselves are being modeled, transformed, subsetted and round-trip engineered. Following the same idea as for business document standards, future work includes the transformation of actual business document instances. Another open research question is the development and integration of matching tools allowing to automatically match different business document standards, hence supporting the alignment of different business document standards. Furthermore, another aspect currently being investigated is the visualization of large models to ensure proper usability of business document modeling tools. Therefore, tools and methods are currently being investigated.

Acknowledgements

The work of Research Studios Austria is funded by the Austrian Federal Ministry of Science and Research. Furthermore, this work has been carried out under the research grant ”Public Private Interoperability” (No. 818639) of the Austrian Research Promotion Agency (FFG).

References

[1] H. Li, “XML and Industrial Standards for Electronic Commerce”, *Knowledge and Information Systems*, vol. 2, 2000, pp. 487-497.

[2] OASIS, *Universal Business Language v2.0*, http://docs.oasis-open.org/ubl/os-UBL-2.0/UBL-2.0.html, Last Visit: May 2009.
[3] Health Level Seven, *HL7*, http://www.hl7.org/, Last Visit: May 2009.
[4] Society for Worldwide Interbank Financial Telecommunication, *SWIFT Standards Family*, http://www.swift.com/, Last Visit: May 2009.
[5] HR-XML Consortium, *HR-XML Standards Family*, http://www.hr-xml.org, Last Visit: May 2009.
[6] A.P. Sheth, "Changing Focus on Interoperability in Information Systems: from System, Syntax, Structure to Semantics," *Interoperating Geographic Information Systems*, vol. 47, Jan. 1999, pp. 5-29.
[7] UN/CEFACT, *Core Components Technical Specification 3.0*, http://www.untmg.org/ccts/spec/3_0, Last Visit: May 2009.
[8] UN/CEFACT, *UN/CEFACT's Core Component Library UN/CCL*, http://www.unece.org/cefact/codesfortrade/codes_index.htm, Last Visit: May 2009.
[9] UN/CEFACT, *UML Profile for Core Components Technical Specification 3.0*, http://www.untmg.org/upcc/spec/3_0, Last Visit: May 2009.
[10] P. Liegl, "Conceptual Business Document Modeling using UN/CEFACT's Core Components," *Proceedings of Sixth Asia-Pacific Conference on Conceptual Modelling (APCCM 2009), January 20-23, Wellington, New Zealand,* 2009, pp. 59-69.
[11] VIENNA Add-In, *An B2B collaboration modeling Add-In for Sparx Systems Enterprise Architect*, http://code.google.com/p/vienna-add-in/, Last Visit: May 2009.
[12] P.A. Bernstein, "Applying Model Management to Classical Meta Data Problems", *Proceedings of First Biennial Conference on Innovative Data Systems Research (CIDR 2003), Online Proceedings, Asilomar, CA, USA*, 2003.
[13] G. Kappel, H. Kargl, G. Kramler, A. Schauerhuber, M. Seidl, M. Strommer, and M. Wimmer, "Matching Metamodels with Semantic Systems - An Experience Report," *Proceedings of Datenbanksysteme in Business, Technologie und Web (BTW 2007), Workshop Proceedings, March 5 - 7, Aachen, Germany*, 2007, pp. 38-52.
[14] UN/CEFACT, *XML Naming and Design Rules 3.0, ODP5*, United Nations Center for Trade Facilitation and Electronic Business, 2008.
[15] T. Bouras, P. Gouvas, and G. Mentzas, "Dynamic Data Mediation in Enterprise Application Integration," *Proceedings of the eChallenges e-2008 Conference, October 22-24, Stockholm, Sweden*, 2006.
[16] W3C, *Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language*, http://www.w3.org/TR/wsdl20/, Last Visit: May 2009.
[17] W3C, *OWL Web Ontology Language*, http://www.w3.org/TR/owl-features/, Last Visit: May 2009.
[18] W3C, *Web Service Semantics - WSDL-S*, http://www.w3.org/Submission/WSDL-S/, Last Visit: May 2009.
[19] M. Nagarajan, K. Verma, A.P. Sheth, J.A. Miller, and J. Lathem, "Semantic Interoperability of Web Services - Challenges and Experiences," *Proceedings of the 2006 IEEE International Conference on Web Services (ICWS 2006), September 18-22, Chicago, Illinois, USA*, 2006, pp. 373-382.
[20] A. Bouras, P. Gouvas, and G. Mentzas, "ENIO: An Enterprise Application Integration Ontology," *18th International Workshop on Database and Expert Systems Applications (DEXA 2007), September 3-7, Regensburg, Germany*, 2007, pp. 419-423.

[i] http://ios.researchstudio.at
[ii] http://www.altova.com