RepXML: storing B2B data using an ebXML Registry

Ivan Bedini[1], Fabrice Bourge[1], Benjamin Nguyen[2]

1: France Telecom
2: PRiSM Laboratory - University of Versailles – France

ivan.bedini@francetelecom.com, fabrice.bourge@francetelecom.com, benjamin.nguyen@prism.uvsq.fr

Abstract

Unlike traditional Electronic Data Interchange (EDI) which mostly applies to long term partnerships and tends to involve only large companies, B2B e-commerce promises to reduce the set-up and operational costs and to provide greater flexibility. However there are still many difficulties at design time when establishing a business collaboration, and to bridge the gap from design time to run time. We focus here on the storage of B2B data in an ebXML registry for reuse. After a brief state-of-the-art, this paper presents a concrete implementation of such a registry with real business users. The experience shows that an ebXML registry provides a good opportunity to register and store many types of e-business components and that it could efficiently be used to simplify part of the set-up of B2B exchanges.

Keywords

B2B, e-commerce, core component, CCTS, ebXML, Open-edi, registry, repository, RepXML

1 Introduction

1.1 Context

When conducting a business relationship with its partners, any company, regardless of its size, seeks to increase its operational efficiency by improving the business processes and lowering costs. One way of reaching this goal is to automate the business processes to gain time (reduced transportation delays and increased speed of processing), and to reduce human intervention, therefore errors. Of course this applies to the operations performed both internally (inside the company) and externally (with other partners); we specifically focus in this paper on communications between companies.

Additionally most companies want to expand their market share, which implies finding new partners or conducting more business relationships. E-commerce is not limited to "commercial" companies. Administrations are also confronted with similar problems in their relationships with companies or even other administrations: they need to provide high quality services to a wide audience, targeting both the private and public domains, while improving their efficiency and reducing their costs.

Since the 1960's, an important effort has been made to try to define standard data formats so that business partners could exchange structured business data via automated means, i.e. directly between computer-supported business applications [2]. Over the years numerous Electronic Data Interchange (EDI) standards have been defined to enable interoperability. To this day, the UN/EDIFACT and ANSI ASC X12 standards are the most widely used. However traditional EDI suffers from barriers such as development and utilisation cost, technology limitations, complicated standardisation processes and critical user mass [10]. As a result, most of the EDI implementations that have been successful only apply to long term partnerships with high volume exchanges, and tend to involve only large companies.

In the mid 1990's, the advent of the internet and its related technologies, in particular XML, has lowered these barriers by reducing the set-up and operational costs, while adding greater flexibility. This new way of doing business between companies is commonly referred to as business-to-business electronic commerce (B2B).

Another major change occurred in 1997 with the publication of the ISO Open-edi Reference Model standard [9]. Open-edi introduces the notion of standard business scenarios and the services needed to support them. Once trading partners commit to an agreed business scenario and use Open-edi conformant implementations, there is no further agreement needed between them. This makes it possible for all types of companies, including SMEs, to rapidly engage in a B2B relationship in a cost effective manner. Overall the Open-edi Reference Model provides two views to describe a business transaction: the Business Operational View (BOV) to address the business needs, and the Functional Service View (FSV) to address information technologies issues. The separation of these two views provides flexibility and durability since the changes on one side do not impact the other side.

The Open-edi vision has influenced several initiatives worldwide, including RosettaNet and the UN/CEFACT's modelling methodology (UMM) which builds upon business semantics when defining B2B collaboration models. Other organizations such as EAN-UCC, SWIFT and TM Forum have contributed to the development of UMM and have aligned their methodologies with it [3]. Even though it is not mandatory, UMM is an integral part of the ebXML framework. It divides a B2B collaboration into two distinct phases: the *design time* phase during which business processes and business documents are defined, and the *run time* phase which executes the business process through collaborating application systems.

In parallel, more and more initiatives also study storage systems, whose goals are the sharing of business artefacts. This is the case for several governmental institutions, standardization organizations and large companies or consortia that look for efficient registry solutions where they can publish and share their defined services, business processes and business documents. Such systems are considered fundamental to increase visibility and availability of business data, to increase information sharing, and to simplify the harmonisation of existing artefacts and their reuse. Currently only the UDDI registry and the ebXML registry/repository standards, both specified by OASIS, address this key issue; their wide adoption would help reduce the existing gap between design time and run time. This article describes an experiment using an ebXML registry/repository.

1.2 Problem definition

When setting up a new collaboration, one is often faced with the hard task of defining the necessary business artefacts. Design time includes several tasks that are (at this time) still performed manually or done in an *ad hoc* manner; therefore this process remains very long, complicated, and somewhat arbitrary.

One remaining difficult task is to find existing artefacts that correspond to specific business needs and, of course, to understand how to reuse them. Within the B2B domain, the implementation of business exchanges is usually based on standard solutions. While this is surely a good way to reduce interoperability problems and to benefit from world wide experiences, it also implies that business analysts must first find existing models based on these standards and then start the collaboration design from these models: the problem of finding, reusing and harmonizing artefacts still remains. Until now it has been a common practice, including among some standardization organizations, to simply publish business data on a web page in directories or even in flat files! As a consequence, discovery is tedious and takes a lot of time, since browsing through search results must be performed manually. Some registry solutions are timidly emerging, but a lot of work still remains to be done before these solutions can truly facilitate and support some automation at design time.

Among the many difficulties associated with design time we believe that it is of utmost importance to first define at semantics, syntactic and structural levels the contents of business documents to be exchanged between business partners. Moreover, we need to be able to reference and store the contents elements in a search-efficient and reusable way to simplify the construction of the business document.

We address these problems by advocating the use of a semi-structured repository to store B2B data, according to the ebXML standard, and also to query it in a more efficient way than before.

1.3 Definition of terms and concepts

Throughout this paper we regularly use some terms and concepts that bear a specific meaning. These are defined as follow:

Business artefact, or simply artefact, describes a generic electronic business information used within B2B collaboration exchanges (e.g.: documents, business processes, web services, messages, company profiles, code lists…).

Business data, or simply data, describes a defined set of electronic business information of a specific type necessary to set up and to establish B2B collaborations. The most commons business data are: business process, message contents, web service description, trading profile and trading agreement.

Business document describes the contents of a message exchanged between parties involved in a B2B collaboration.

Business component describes a building block of conceptually related elementary information, similarly to the UML class construct. The aggregation of one or more business components constitutes the content of a business document. (e.g.: address, person, organization, contact, financial account, product, etc…).

Business analyst describes all users responsible for business collaboration modelling and design.

Business expert describes all users with a good knowledge of business collaboration design and authors of harmonization tasks for standardization.

Design time covers all necessary tasks for modelling and for setting up the execution of B2B collaborations. This phase involves the business process specification, the partner profile definition, the trading partner contract establishment, the business document conception and the exchanged message integration (or mapping) to the existing information system. Design time also includes the discovery and retrieval of existing business artefacts.

Run time covers the execution time of business exchanges from beginning to their termination. (i.e., business processes execution, messages exchange and dynamic services discovery).

Note: throughout this paper the term B2B covers common problematic to the whole e-business environment; therefore it also covers A2A, B2A and to some extent B2C and A2C (where A stands for administration, B for business and C for consumer).

1.4 Goal

In this paper we provide a solution that references and stores UN/CEFACT Core Component-based artefacts in an ebXML Registry. These artefacts are assembled to build the exchanged business documents. We show that by attaching the appropriate services to the registry, it is possible to highly simplify and encourage the reusability of these artefacts. Our contributions in this paper are of two different sorts. First of all, we provide a survey of related B2B projects. Secondly, we describe the RepXML project, insisting on its architecture. Finally we provide the direction that future works and researches must investigate to improve the adoption of e-business registries and the definition of interoperable and more flexible business data. Our assumptions are based on the research experience that we accomplished during the implementation of the RepXML prototype.

1.5 Outline

This paper is organised as follows. Section 2 provides a survey of related B2B initiatives with a short analysis of different approaches to management of the structure and semantics of business documents, and business registries. Sections 3 and 4 furnish the RepXML project background and provide feedback on the conducted experimentation. We present in Section 5 the lessons learnt from the RepXML experience, which have supplied us with important elements on which to base future work.

2 Overview of related projects

In this section, we first survey some standards, major B2B solutions or implementations, and e-Government initiatives that within the past few years have proved relevant in data definition and data sharing within the B2B domain. Based on an analysis of all these we propose a categorization of the different approaches to data definition.

2.1 Standards

ebXML Core Component Technical Specification

The Core Components Technical Specification (CCTS - ISO/TC 15000-5) [19] constitutes Part 8 of the ebXML framework. This specification describes a way to develop a set of semantics building blocks for the creation of relevant business vocabularies. These building blocks are either context neutral Core Components (CC) or context specific Business Information Entities (BIE). UN/CEFACT is in the process of providing a freely accessible core component library [20] which will be stored and maintained in an ebXML compliant registry [21].

In version 2.01 of the specification which was used for RepXML, there are four categories of Core Components: basic CC, aggregate CC, association CC, and CC type. Core Components are conceptual models that form the basis for Business Information Entities; Business Information Entities are always derived from their source Core Component. There are three categories of Business Information Entities: basic, aggregate and association. The Business Information Entities are used to create the messages exchanged between partners.

Figure 1 shows the relationships between Core Components and Business Information Entities concepts (dotted concepts and links in the figure are not specified by the CCTS but only informative).

The separation between context neutral components and context specific components provides a great advantage from the reusability point of view. RepXML implements this specification.

The CCTS provides a technology neutral model. Associated with it is the XML Naming and Design Rules specification [22] that provides a set of rules for transforming compliant business components models to XML schemas.

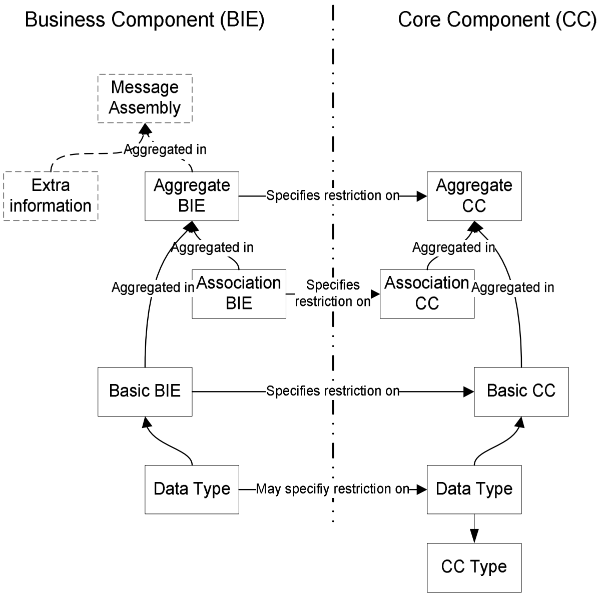


Figure 1 – CCTS main concepts and relationship between Core Components and Business Information Entities

RosettaNet

RosettaNet is a not-for-profit organization founded in 1998 by a group of companies in the high tech sector that promotes collaborative commerce and develops universal standards for the global supply chain [17]. The RosettaNet standards prescribe how to implement collaborative business processes between supply-chain trading partners using networked applications. These specifications are focused on the business processes defined in the Partner Interface Processes (PIP), and also include the business data definitions and technical elements for interoperability and communication, e.g., XML schemas for business documents.

The specified artefacts are published on the web site in the PIP Directory which is a simple organized collection of web links to the Partner Interface Processes specifications. It includes a business document with the semantics and a business process with the choreography of the message dialog. There also exist RosettaNet dictionaries to provide a common set of properties for the Partner Interface Processes.

Even though the RosettaNet standard is being adopted in other business domains, it does not seem to provide any generic model or rules for reuse, therefore limiting its wide use in specific collaborations. Furthermore this prevents a solution such as RepXML to adopt the RosettaNet standard.

UBL

The Universal Business Language (UBL) [15] is the product of an international effort to define a standard library of electronic XML business documents. It is developed in the UBL OASIS Technical Committee with participation from a variety of industrial data standards organizations. The definition of business data is based on the ebXML Core Component Technical Specification [19] constructs. The main work done in this group is to define and harmonize the semantics of the invoice and order documents. The UBL Technical Committee does not provide any type of registry where data can be queried and reused; on the contrary all XML schemas and business processes definitions are delivered directly within the specification package. An interesting part of this work deals with a detailed set of structured and unstructured data types for object instances that can be reused in other contexts and simply extended.

As the UBL library follows the CCTS constructs, a test has confirmed that the contained components can be registered within RepXML.

XBRL

XBRL (eXtensible Business Reporting Language) [24], developed by an international not-for-profit consortium, is a language geared towards electronic communication of business and financial data. It is becoming a standard means of exchanging information between businesses. It specifies a set of common semantics and structures for financial documents with added specific functions, called taxonomies, which enable unique identifying tags to be applied to financial data items, such as "net profit". The XBRL taxonomies are an approach that can be considered orthogonal to the core components but do not present a cross-area scope and reuse.

The standard XBRL taxonomies are published via a web site as a list of documents divided by country. The XBRL consortium is leading to the proposal of new, non-standard roles having common and useful semantics to be published within the XBRL Link Role Registry (LRR) that will provide structured information about their purpose, usage, and any intended impact on XBRL instance validation.

UNIFI (ISO 20022)

ISO has developed a framework that provides the financial industry area with a way to define messages in a standardized XML syntax; this is called the UNIversal Financial Industry message scheme (UNIFI) and corresponds to International Standard ISO 20022 [10]. The UNIFI framework comprises a development methodology, a registration process and a central repository where approved contents are stored maintained by SWIFT (Society for Worldwide Interbank Financial Telecommunication).

ebXML Registry Repository specifications

The **ebXML Registry Services and Protocols specification** [13] (ebRS V2.0 is ISO/TC 15000-4) provides a set of services that enables the creation of content management systems for secure and federated information sharing.

The shared information is maintained as items in a repository, while the registry maintains the corresponding metadata. The registry content is managed through the standard Life Cycle Manager interface (LCM), which is the interface responsible for all object lifecycle management requests, and the standard Query Manager interface (QM), which is the interface responsible for handling all query requests.

In May 2005 version 3.0 of this specification was approved as an OASIS standard. Compared to previous versions, it provides enhancements on secure information management capabilities, content management capabilities, content versioning, storage of complex queries in the registry in parameterized form, content-based notification capability, etc.

The **ebXML Registry Information Model specification** [12] (ebRIM V2.0 is ISO/TC 15000-3) provides information on the type of metadata and contents that are stored in an ebXML registry as well as the relationships among metadata classes. It predefines lists of extensible canonical objects, attributes, associations, types, etc. in a generic manner and defines how stored objects are organized and classified. It does not specifically address the storage of any particular business data.

Version 3.0 of this specification was also approved as an OASIS standard in May 2005. It appears that it is not fully backward compatible with previous versions as the information model has evolved (e.g., the "RegistryEntry" class has been removed; the "Person" class has been added, etc.). This impedes federation between a registry based on previous ebRIM versions and one based on ebRIM version 3.0.

UDDI

The Universal Description, Discovery, and Integration (UDDI) is a platform-independent, XML-based registry for worldwide business services to list themselves on the Internet. UDDI is an open industry initiative (sponsored by OASIS) enabling businesses to publish service listings, discover each other and define how the services or software applications interact over the Internet. UDDI version 3 provides a rich description of web services and robust queries for locating web services. Meta-data includes standard white, yellow, and green pages to describe web services, operational information, complex categorization, WSDL support, multiple overview documents, and extensibility. UDDI has succeeded at becoming a central component of web service architectures. In version 3, a notification interface is also available. For the time being, most implementations are based on version 2 and only include basic identification and categorization information for web services.

Note: the ebXML registry/repository and UDDI specifications have been developed in parallel by OASIS within the past few years. As shown in the registry standards comparison matrix in [26], these two specifications have several concepts in common, but while the latter is specialized on services information management, the former allows a larger vision of extensible e-business manageable artefacts and useful application services.

2.2 e-business registry implementations

More and more often the e-business world asks for an advanced storage system for managing and sharing business data. Hereafter we shortly present four implementations of registries that are similar to RepXML, three of which are based on an ebXML Registry.

REMKO

Initiated in 2001 [1] by the Korea Institute for Electronic Commerce (KIEC), the ebXML Registry & Repository in Korea (REMKO) [16] provides and maintains Korea EDIFACT Committee (KEC)-approved standard electronic messages, code lists of electronic messages (XML and EDI), company profiles, ebXML related contents, etc. Its purpose is to promote ebXML, to enable interoperability and electronic collaboration across industries in Korea, to share information with federated industrial sector registries (e.g., in the steel industry), and to support small and medium companies with global trade. In 2004 REMKO was linked to the Basic Semantic Register (BSR) to interconnect various data elements semantically across multiple industries.

LomakeFi

During 2003 Republica ran a project aimed at producing electronic forms for the Finnish Government based on existing paper forms. The project team set up and maintained a registry and created the LomakeFi Form Assembler tool to assemble documents from the registry information entities in order to generate forms and produce XML schemas. The project used the ebXML Core Component approach and relied on an ebXML

[1] At the time where we started the RepXML project we were not aware about this initiative and of any public communication from the Korea Institute for Electronic Commerce

registry to store information. A pilot with Finnish public administration personnel was started in September 2003.

The UN/CEFACT Registry Implementation Specification

In 2004 the UN/CEFACT Information Content Management Group started a project to develop the future UN/CEFACT Registry where the UN/CEFACT cross-sector business artefacts will be published and made available to all. It will provide reusable contents mainly in the form of Core Components (CC) and Business Processes. The UN/CEFACT Registry will also be used for storing, managing and distributing UN/CEFACT deliverables. Furthermore it is assumed and foreseen that there will eventually be many ebXML registries across the world, for example to satisfy the specific needs of a geographical region, a business community, etc. The UN/CEFACT Registry adopts the ebXML registry federation service to foster interoperability in e-business exchanges.

The UN/CEFACT Registry Implementation Specification [21] specifies this global directory. It is largely based on the RepXML experience. Version 1 of this specification which is done under the editorship of France Telecom, should be available sometime this year, and the actual UN/CEFACT Registry will be implemented after a proof of concept phase.

Swift Financial Directory Repository

The repository for UNIFI components seen above is maintained by SWIFT [18]. It is made of two parts: the "data dictionary" that contains all the components used in models and message formats, and the "business process catalogue" that contains business models, transactions and messages. Both, the Dictionary and the Catalogue form the Repository, i.e., a well organized central repository where all the defined artefacts are stored and publicly accessible. This is an interesting initiative that allows the interested parties to simply look for and retrieve the minimum set of information required in a financial B2B process.

2.3 e-Government initiatives

Canadian e-government project

The Ontario Gateway is a major e-Government project currently underway in Canada. It aims at providing the government of Ontario ministries with an ebXML registry-based message brokering and document publishing hub to securely exchange business information with counterparts in other Canadian jurisdictions. It uses the CSDML (Canadian Service Description Markup Language) ontology which describes the government knowledge domain in the form of entities, properties, relationships, taxonomies and rules.

Federal Enterprise Architecture – The Data Reference Model

The Data Reference Model (DRM) [23] is a part of the Federal Enterprise Architecture (FEA) which is an initiative of the United States' government. The FEA aims at defining a set of interrelated reference models for «conducting and facilitate cross-agency analysis and the identification of duplicative investments, gaps, and opportunities for collaboration within and across Federal Agencies» (from [23]).

As a reference model, the DRM is presented as an abstract framework from which concrete implementations may be derived. The DRM provides a flexible and standards-based approach to enable data discovery, reuse, harmonization and sharing. Its scope is broad, as it may be applied within a single agency, within a Community of Interest, or across Communities of Interest and it is independent from the implemented methodologies and technologies. On of its interesting aspects is its approach on semantics interoperability: The model presented in the DRM introduces the semantic description as an important part of the data and the data context construction and this approach is sufficiently open to let future implementations integrate semantics concepts, such as RDF or OWL, directly within the data definition. This feature should facilitate the problem of the discovery, integration and harmonization tasks.

2.4 Different approaches to data definition

The constraints that companies are facing when defining B2B data are manifold. They depend on the business collaboration and therefore each collaboration must be evaluated individually. As seen above, there are currently several solutions addressing the problem of defining business data within the B2B domain. These show some analogies. We can distinguish three mains facets when dealing with the problem of data definition:

A) Use of standards – this is the most commonly implemented facet. This can either refer to **specific area** standards or **generic area** standards. A specific area standard focuses on a particular business area or a business process and defines artefacts accordingly. This is the case, for example, with the RosettaNet Business Dictionary [17] for the supply chain area, with the SWIFT Financial Dictionary [18] for financial interchanges area, with the XBRL Taxonomy [24] for the financial business reporting area, and UBL, that focuses on the business data definition for the invoice and order specific business processes area. On the other hand a generic area standard tries to maintain a cross-area scope, such as the ebXML Core Components Technical Specification [19] that provides a library with data harmonized across different business areas [20]. The use of a standard approach follows a strict definition of semantics, syntax and structure of business data with their associated business processes. The advantage of using standards is the higher chances to reach interoperability when implementing business collaborations and an improved ability to consider a larger vision of modelled data. On the contrary there is a lack of extensibility and flexibility that prevents companies from defining new types of collaborations that have not yet been considered within the standard.

B) Use of ad hoc solutions – in some cases content definition is left at application design level when the data model is not the priority. This is often the case in occasional partnership with point-to-point exchanges that do not require a large degree of openness and evolution. The main advantage in adopting this approach is the flexibility that it provides. It is completely independent from semantics, structure, syntax and implementation of a specific solution. An agreement between the parties involved is enough to conduct the collaboration. One of the drawbacks is the low degree of integration of new partners in collaborations.

C) Semantic approach – some solutions are based on semantics technologies [5] [23] even though none have yet been implemented for B2B applications. Here the ontology of the business documents could be enriched with semantics definitions. This approach seems to provide an interesting flexibility when matching similar semantics concepts and consequently could simplify integration and harmonization and improve interoperability. The inconvenience is the lack of reliable matching tools and the introduction of a new complex task which is the definition of the semantics description and relationships of data.

In conclusion none of these approaches provides on its own a complete and adequate B2B solution. None of the surveyed standard, initiative or implemented solutions provides the necessary degree of interoperability, flexibility and extensibility required by B2B applications. The current focus is clearly on the sole interoperability issue as if it was the only possible choice for defining business data in B2B solutions. However, based on their complementary qualities it seems that a solution could implement all three aspects at the same time, therefore improving flexibility and reusability and facilitating harmonization.

3 Project background

Before entering into the details of RepXML (see next section) we provide here an overview of the project, its positioning with respect to data definition approaches as well as to other solutions, and the work environment.

3.1 Goal and objectives

Started at the end of 2004, RepXML's goal is to simplify and encourage B2B exchanges among French companies or with administrations by providing a common registry accessible on the internet where one can easily find and reuse business components [8]. The business components handled by RepXML can be composed of different types of elements (ebXML-based business elements, XML Schemas, metadata information, other related business documents, etc.). In the remainder of this paper these business components are referred to as Business Specifications.

RepXML can be accessed via a Web interface or via a client software application called the Connector which serves as Business Specification editor. As it is necessary to start by populating the registry, an important focus is to help business analysts edit and submit new Business Specifications to RepXML. The submitted Business Specifications are then examined by an expert who performs some harmonization with respect to already approved Business Specifications before approval.

More specifically, the objectives of the RepXML project are as follows:

- implement a repository that registers and stores Business Specifications,
- develop a web interface, including a public area and a restricted access area reserved to specific types of users,
- define and implement a procedure that enables users from business sectors (e.g., business analysts) to submit, validate, retrieve and re-use Business Specifications,
- develop a client application (the "Connector") that allows users to edit new Business Specifications off-line,
- develop the interface between the Connector and the repository server,
- conduct a field trial with real business users (from the Edifrance community) to get feed-back.

3.2 Positioning

Data definition approach

Fundamentally RepXML adopts approach A – use of standards – as defined above (Section 2.4), but it also leaves users with the possibility to define new business components even if none of the existing context neutral components matches the required concept. This implies that a business component may not always be derived from an existing context neutral core component from the official library [20], thus conflicting with CCTS [19]. After a first feedback from experimenting users it was decided to keep this (loose extensibility from core components) as a valuable RepXML feature. And so the RepXML approach can be categorized as a mix between approaches A and B (standards and ad hoc). The integration of the semantic approach is considered for future improvement of the solution.

Targeted advantages

In addition to what is provided in other solutions, RepXML supports the whole chain of business component design. It provides an editing tool for the considered business components, and public interfaces for accessing and managing as simply as possible the specific registry content. It also allows users to export the discovered information in a common, formal language or machine processable format so as to increase reusability of business components when building business documents for B2B.

3.3 Work environment

The registry plays a fundamental role in RepXML's goal to provide a means to reference and share business data in a place that is well known, widely accessible, reliable, and easy to use. The registry is the container, and the business data is the content. This content needs to have structure and semantics that are relevant to the business needs and that enables reuse among different fields. Furthermore the contents must be compatible with the handling capabilities of the registry, i.e. XML.

The RepXML project being oriented towards the implementation of a concrete service prototype with real users, time, resources and budget constraints are of course associated with it. In this context, we had to make important choices early on that would have a major impact during the course of the project. Based on prior knowledge on ebXML acquired during previous projects at France Telecom R&D, we realised that the ebXML registry/repository specification constituted a good basis to reach the project's goal. Thus it was decided to use the FreebXML Registry software which is the most complete open source implementation of the ebXML Registry specifications ([12],[13]) developed by OASIS. Regarding the contents it was decided that part of the Business Specifications' contents would have to comply with the UN/CEFACT Core Component Technical Specification (CCTS) [19].

Those choices were not made on the basis of a thorough study of the best approach to the problem we were facing. However, based on our own knowledge and experience, we believed that those choices yielded a very good chance to succeed and that it was of great interest to investigate their use in a real case. Furthermore, these choices were compatible with the B2B environment and the desire to use open standards for interoperability issues.

3.4 ebXML framework

It is not the purpose of this paper to educate the reader on the entire set of the ebXML specifications. However it is important to provide an overview of the framework that RepXML relies on, and to explain their interest in our experience.

The ebXML framework defines a set of rules that business communities or professional sectors can use to register their data and their business practices. Companies or administrations can use this business data or practices through the ebXML standard infrastructure, which is based on a registry and reliable and secure messaging services.

First issued in May 2001, the different ebXML specifications have not evolved at the same pace and consequently have not reached the same maturity state [4]. A lot of work still needs to be done to attain significant growth in the number of companies and administrations using ebXML.

This lack of maturity is particularly true in the development of business libraries where one could find a shared set of building blocks representing the general types of business data in use in their business field. Some of the standardisation work currently done at UN/CEFACT and OASIS aims at filling this gap much needed by companies and administrations.

3.5 Partners

Three partners are involved in the RepXML project:

France Telecom – For several years now, France Telecom R&D Division has been studying closely the ebXML framework through several internal projects. France Telecom is responsible for the server side of RepXML and for the field trial.

SRCI – Based in France SRCI is a small-sized software company specialised in EDI and process oriented interoperability. SRCI is responsible for the client side of RepXML.

Edifrance – Created in 1990 Edifrance is a not-for-profit organisation that promotes and develops ICT (Information Communication Technology) among French companies and administrations. It numbers over 150 direct members and thousands of indirect members. The RepXML field trial was conducted among members of Edifrance.

4 The RepXML experience

In the following two subsections we present the RepXML experience use cases and users' roles. In subsection three we describe the architecture with a focus on developed modules and successively the main problems that we encountered. In the fourth subsection we provide some points of interest of the field trial, before concluding with considerations on the experience.

4.1 Users

There are five categories of users interacting with RepXML:

- Author: a person that creates a new Business Specification (e.g. for the needs of a particular business community);
- Submitter: a person or an organisation that submits a newly created Business Specification;
- Expert: a person or a group of persons from a validation authority that approves the new proposed Business Specifications (e.g. an expert from the Harmonization Working Group at Edifrance)
- Guest: a person from a company or an organisation that retrieves part of an approved Business Specification from the registry;
- Administrator: a person from the organisation in charge of managing and maintaining the server.

4.2 Use cases

There are three main use cases of interest for this paper:

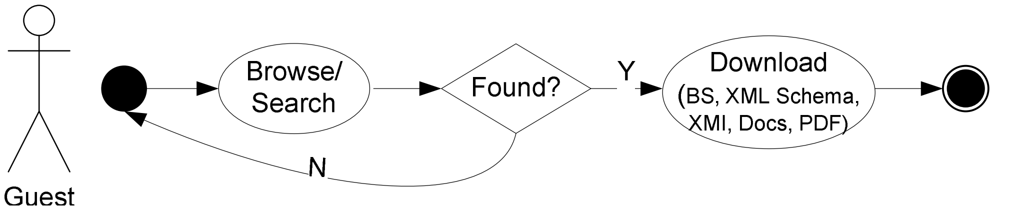


Figure 2 - Discovery use case

1) Discovery use case (Figure 2) – A guest user browses the RepXML web site to look for existing business documents, business collaborations and/or any other information relevant to his business activity. When found the information can be downloaded directly in a format that is appropriate for reuse (e.g., XML Schema, UML class diagrams in XMI…). Let us stress that this is a big improvement compared to UDDI "plain text" search.

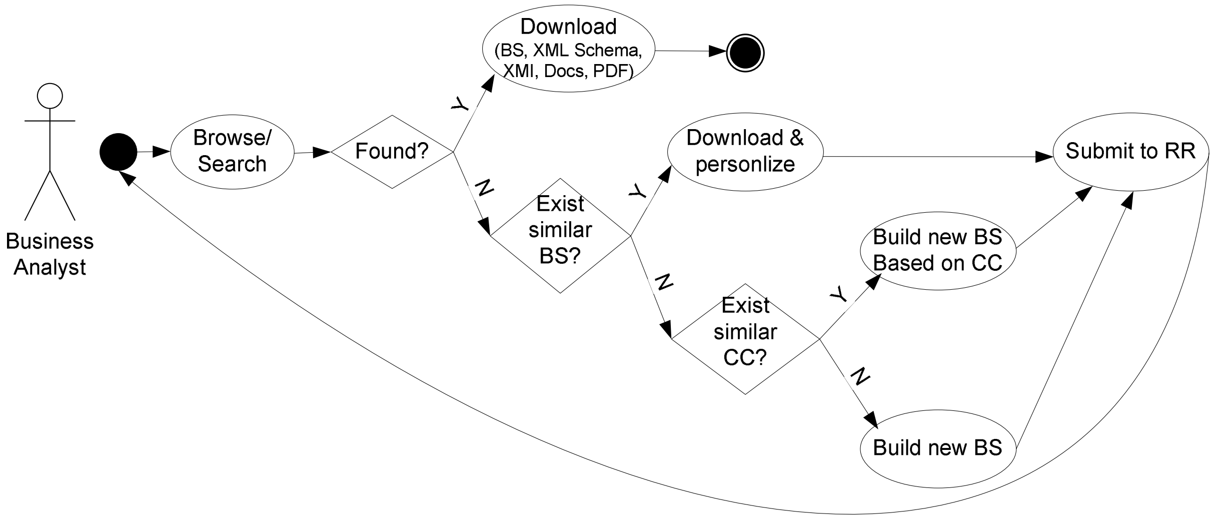


Figure 3 - Design/submission use case

2) Design/submission use case (Figure 3) – Before establishing a new business collaboration, a business analyst must define the business documents requirements, structure and semantics. As in the discovery use case above, he will first search RepXML through the Connector for reusable pieces of information. If none of these fully match the business requirements, then he can check if similar Business Specifications exist. If this is the case, he can download the selected Business Specification into the editor, personalize it and submit the new Business Specification to the registry. If this is not the case, the search can be cascaded down to the standard context neutral Core Components level. Depending on the result of the search, the business analyst can either create a new Business Specification based on relevant Core Components or create a new Business Specification from scratch, and submit it to the registry.

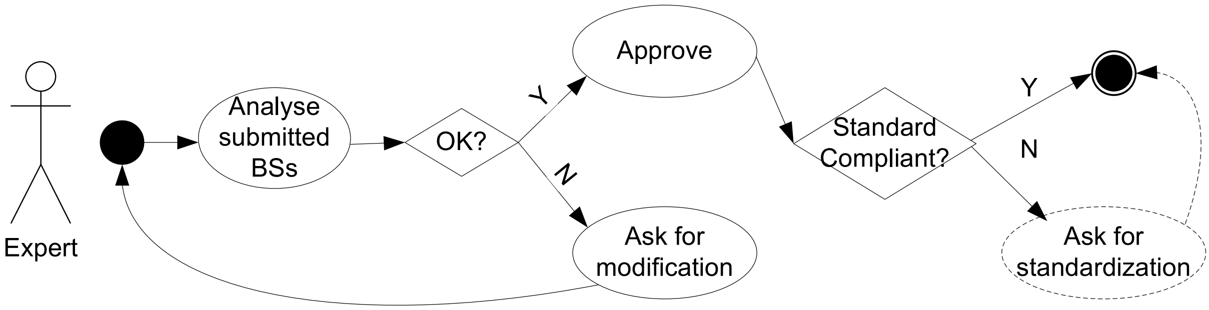


Figure 4 - Approval use case

3) Approval use case (Figure 4) – An expert from a validation authority can query the registry to see if a new Business Specification has been submitted. After analysis, if it is compliant with the standard and does not violate any semantics, the expert can either approve or reject the Business Specification, or ask for some modifications by the submitter before giving his approval. In case of approval, even if the Business Specification is not based on existing standard Core Components, the expert should prepare a submission to the standard authority to ask for the extension of the controlled vocabulary library. (As RepXML allows the construction of semantically incompliant business components, dotted task and link are not covered by RepXML, but they should be the natural evolution of the use case).

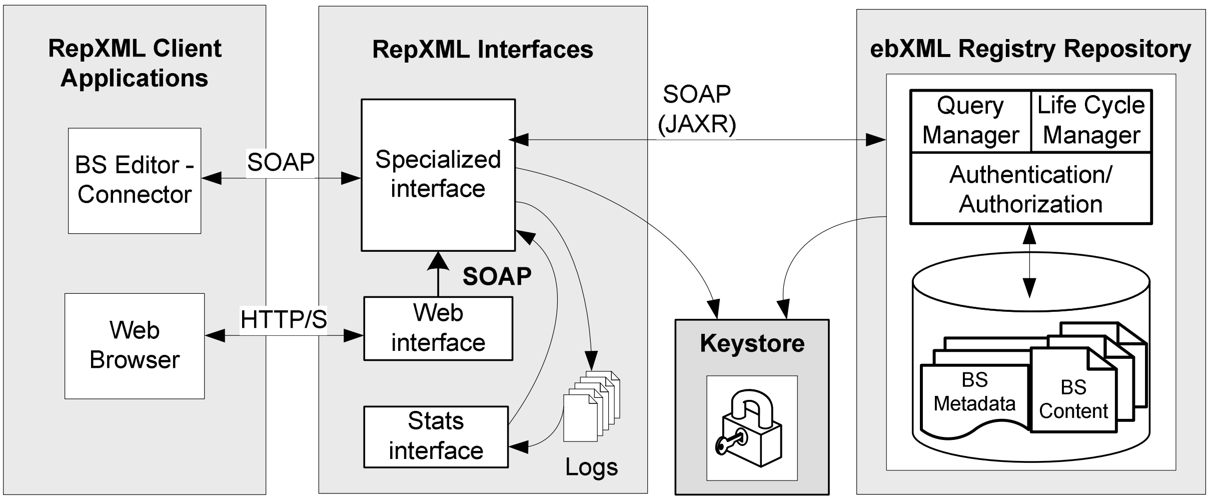


Figure 5 - RepXML architecture overview

4.3 Architecture

RepXML is built on a modular architecture providing a full set of software tools and applications that enable users to discover, build, store and manage Business Specifications. The overall architecture, illustrated in Figure 5, shows the four main modules. In this project, a few modules were reused (Keystore module and ebXML Registry), but we had to develop all the other ones.

The Keystore module contains the users' public and private keys and is mainly used by the registry for authentication. The ebXML compliant registry is an open source effort named freebXML Registry Repository [29].

RepXML Client Applications

A user can communicate with the ebXML registry using both the Business Specification Editor, called also Connector, or a web browser.

The lack of existing tools able to implement and build Business Specifications, compliant with the Core Component Technical Specification, has brought us to develop a dedicated tool. The connector is a stand-alone application, developed in .NET technology, which can be freely downloaded [28] and installed by users on their personal computer (currently only for Microsoft Windows operating system). Through a set of predefined Web Services the connector is able to communicate with the RepXML interface and thus with the ebXML registry/repository.

The connector allows business analysts to retrieve and synchronize the standard library of existing core components, to build Business Specifications locally and to submit these to the ebXML registry via the RepXML interface. Before creating a Business Specification, one has the possibility to perform a search on registered Business Specifications, either locally in the Connector or in the registry always. This search is conducted via the remote RepXML interface.

The RepXML web site also allows users to access the registry contents and submit a Business Specification, but in this case the Business Specification must be submitted directly in the defined XML exchange format (detailed information about the RepXML exchange format is developed in section 4.4 below). Thanks to its dedicated user-friendly interface the connector improves the ability to develop Business Information Entities compliant to the Core Component Technical Specification much faster.

RepXML Interfaces

The RepXML interfaces module considerably facilitates the communication between client applications and the ebXML Registry.

The ebXML registry specification [13] defines two standard interfaces to manage and query the repository content. The implemented freebXML registry provides these two interfaces which are accessible by every client application. Nevertheless the specific contents defined in RepXML require an important knowledge about how artefacts are stored within the registry and how to retrieve and manage them. For this reason a specialized java application interface, implementing the Java API for XML Registries (JAXR), has been developed upon the standard ebRS interfaces to hide the registry queries and management complexity for structured contents.

The **specialized interface** represents a bridge between the standard interfaces and provides the specific mapping from the core component information model (the source model) and the ebXML Registry Information Model. Just to give an idea about what services this interface offers let us recall that the creation of a single business specification in the registry requires on average 50 registry objects; this means that a single call to the create Business Specification method of the specialized interface creates in fact 50 registry objects. All methods developed may be used by a RepXML client application, using a set of predefined Web Services (defined in SOAP) to query, create, modify, delete, update and administrate the registered Business Specifications and users.

A web site (www.RepXML.org) provides the RepXML **Web interface**. It allows guest users (development teams, IS designers…) to access and consult the RepXML contents, and to retrieve Business Specifications in different formats. It currently supports XML, UML/XMI, XSD (compliant to the UN/CEFACT XML Naming and Design Rule specification [22]) and PDF. Furthermore, by using this interface, the “validation” authorities may approve or reject submitted Business Specifications. The RepXML administrator can manage registered users through this interface also.

The **stats interface** provides real time statistics on registered objects submitted by each participating organization and also interprets logs file that detail user activity. As RepXML is only a prototype, this interface has been really useful during the experimentation phase to understand users’ needs and problems with the application.

JAXR

RepXML is developed as a client application for the ebXML Registry implementing the SUN Java API for XML Registries interface (JAXR), which is the same interface that can be used when interfacing a UDDI client application with a UDDI registry. In fact, as the two XML registries – ebXML registry and UDDI – manage similar concepts (e.g., identification, association, categorization, service, registry object, etc.), JAXR provides a full mapping to both specifications [27]. We feel that another strong aspect of this approach is that RepXML could also be extended to be used with UDDI registries.

RepXML Business Specification

A RepXML Business Specification is the representation of a business requirement that a business community wants to submit. After structural, mandatory attributes and several other CCTS rules validations, the Business Specification is stored in the registry. A Business Specification represents a set of information that is exchanged between the Connector and the RepXML interface for a submission to the registry. This set of information enables users to qualify business components such as the Aggregate Business Information Entity (ABIE) as defined into CCTS [19]. Figure 6 below shows an example of Person and Address ABIEs with theirs BBIEs and Residence ASBIE (see glossary for definition of terms).

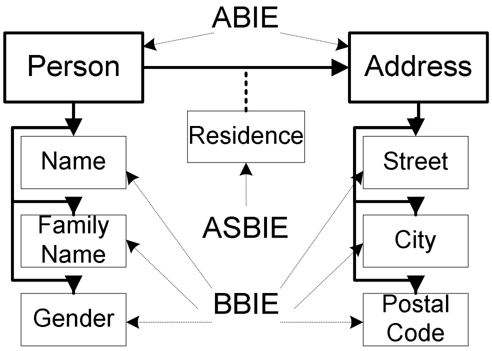


Figure 6 - Example of simple Person and Address ABIEs

In addition to the ABIE semantics, structure and syntax information, the Business Specification also collects information on its Basic BIEs (representing the attributes) along with their type (e.g., string, date), and on associations with others Business Specifications, as well as on all administrative information (like version number, author, responsible organization, related documents, etc…).

Figure 7 below shows the structure of a Business Specification.

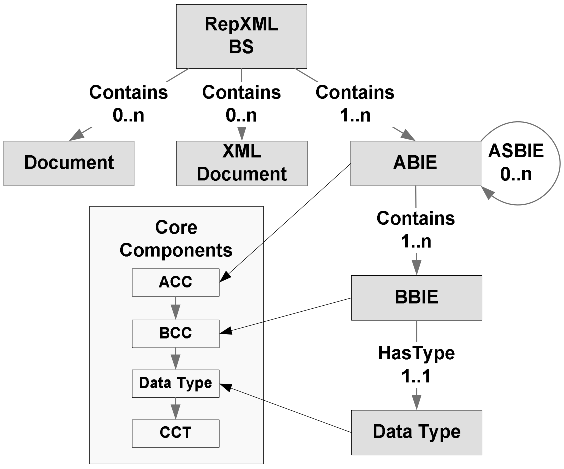


Figure 7 - RepXML Business Specification definition

Business Specification instances are exchanged between the Connector and the RepXML specialised interface as a SOAP message with attachment, where the attachment is the corresponding XML instance of the RepXML business specification XML schema.

Registry profile for Business Specifications

The ebXML Registry specifications do not specify any type of business data registry object management, but only a set of predefined generic objects and queries. In order to implement a registry solution, the profiling task must complete the architecture definition. Concerning ebRIM profiling, the task is similar to the well-known definition of tables and relationships in a typical database system. For further information a detailed profiling template and guidelines are available in the deployment profile template for ebXML Registry document [14], of which we are also the responsible editors.

4.4 Problems and solutions

In this section we present the main problems that we encountered during the development of RepXML both at core and technical level.

CCTS compliance

At the moment the UN/CEFACT standard library [20] is not used widely enough to be considered a generic adoption of a CCTS-compliant solution. This is because each BIE must be based on an existing core component (CC). This rule may significantly limit the creation of new Business Specifications. For this reason in our experience we do not enforce this rule but only suggest its use whenever possible. Nevertheless the structure of the component being built and the naming rules are respected.

It seems to us that the above-mentioned limitation is a major problem since the CCTS relies on the component name to establish the link between a context specific component (BIE) and a context neutral component (CC). . To avoid this limitation, derivation should not be based on names but rather should be based on a conceptual representation of components.

Exchange format

When submitting business components information to the registry and retrieving them, it is necessary to have a shared exchange format between the client application and RepXML, or the registry. This exchange format must be able to carry all the information required by the application and at the same time it should be easily implementable. The CCTS document details the UML class diagram for storing information, but does not detail how business components’ information is exchanged between applications. Based on our experience it would be more useful, for example, to also have a standard XML schema.

Thus, for RepXML we have defined an *ad hoc* XML schema that fills the RepXML application requirements and that could be adopted by future implementations that require exchanges of CCTS compliant information.

Business Information Entities storage

The main problem here is how to store Business Information Entities within the registry. The UN/CEFACT Core Component specification provides detailed UML class diagrams for managing Core Components and the ebXML Registry provides a specific information model, ebRIM. Some UML defined classes can be mapped to pre-existing ebRIM concepts in an ad-hoc and artificial way.

There are two common approaches: one is to create a unique registry object with some metadata and to store the Business Specification as a repository item; the other is to split the Business Specification into several registry objects, one for each main concept. The former shows an extremely simple management of registry content but does not offer great extensibility and reusability, while the latter complicates the development but offers a broader use of registered information.

In RepXML we have followed the second approach by adopting and improving the LomakeFi experience [7]. It represents a good compromise between stored information to maintain compliance with the standard and a reduction of managed information of business components to optimise query performances.

To improve the interoperability between distributed registry implementations, it is important that business data be as homogenous as possible, if possible totally. Our experience acquired on this matter allows us to contribute to define a compliant mapping of the CCTS meta-model to the registry information model. This experience is currently used in defining the UN/CEFACT Registry Implementation Specification [21].

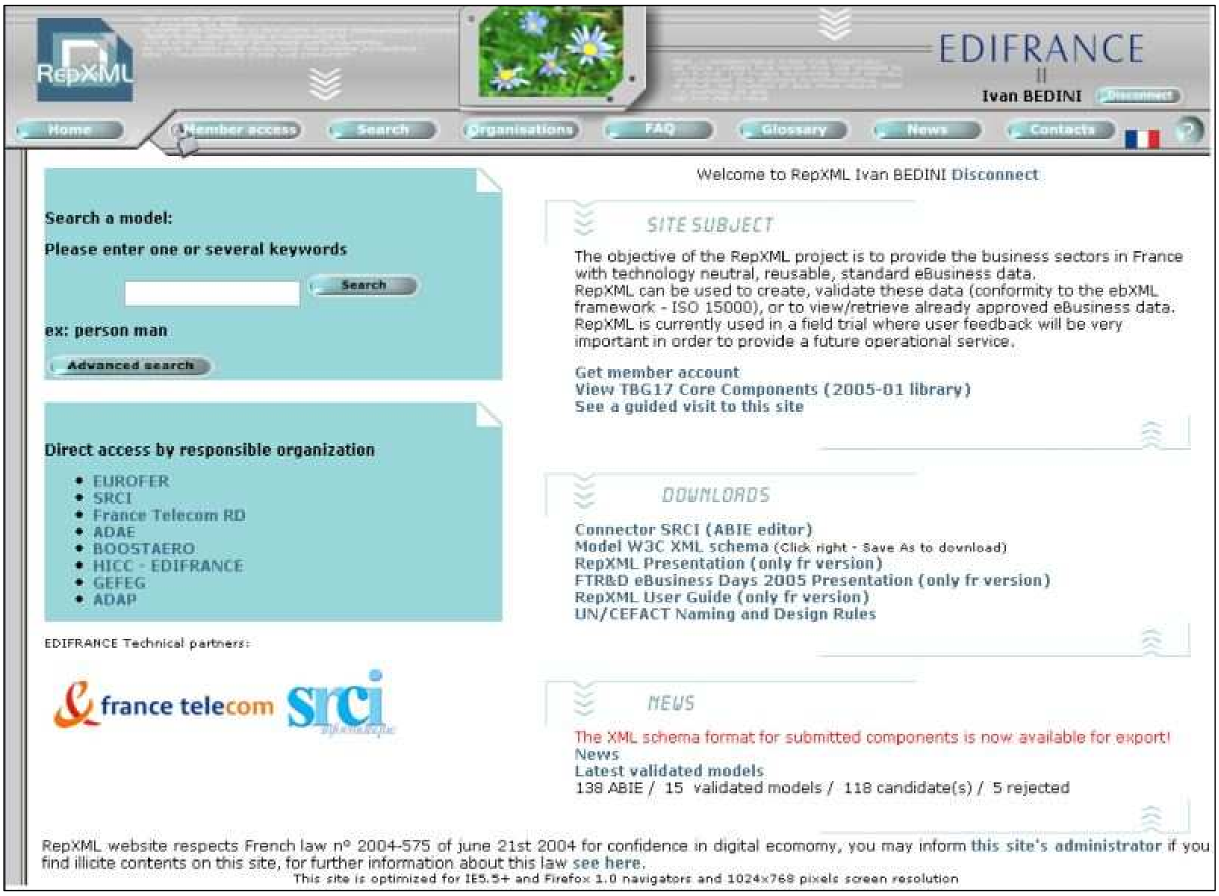
Multilingualism

The multilingualism support appears to be a real challenge when developing international interoperable business documents; in RepXML some issues remain partially unsolved.

RepXML mainly targets French speaking business users and, as a consequence it supports both French and English to define and display Business Specifications. The major problem lies in the semantic correspondence between the different languages used and in the multilingual management of stored artefacts. To solve the latter problem we straightforwardly used the capability of the registry to store most metadata in several languages, e.g. name and description. The binding between English and French is left to the analysts when creating the Business Specification.

4.5 Experimentation

The RepXML application has been used in a six-month field trial that started in early 2005 with Edifrance members. As we write these lines, the web site is still in use even though the field trial is



Search a model:

Please enter one or several keywords

ex: person man

Direct access by responsible organization

- EUROFER
- SRCI
- France Telecom RD
- ADAE
- BOOSTAERO
- HICC - EDIFRANCE
- GEFEG
- ADAP

EDIFRANCE Technical partners:

france telecom srci

Welcome to RepXML Ivan BEDINI Disconnect

The objective of the RepXML project is to provide the business sectors in France with technology neutral, reusable, standard eBusiness data.
RepXML can be used to create, validate these data (conformity to the ebXML framework - ISO 15000), or to view/retrieve already approved eBusiness data.
RepXML is currently used in a field trial where user feedback will be very important in order to provide a future operational service.

Get member account
View TBG17 Core Components (2005-01 library)
See a guided visit to this site

Connector SRCI (ABIE editor)
Model W3C XML schema (Click right - Save As to download)
RepXML Presentation (only fr version)
FTR&D eBusiness Days 2005 Presentation (only fr version)
RepXML User Guide (only fr version)
UN/CEFACT Naming and Design Rules

The XML schema format for submitted components is now available for export!
News
Latest validated models
138 ABIE / 15 validated models / 118 candidate(s) / 5 rejected

RepXML website respects French law n° 2004-575 of june 21st 2004 for confidence in digital ecomomy, you may inform this site's administrator if you find illicite contents on this site, for further information about this law see here.
This site is optimized for IE5.5+ and Firefox 1.0 navigators and 1024x768 pixels screen resolution

Figure 8 - RepXML home page (www.RepXML.org)

finished. Figure 8 below provides a snapshot of the RepXML home page.

Field trial figures

During the field trial 138 Business Specifications were submitted to the registry, of which 15 have been approved by the validation authority and 5 have been rejected. A total of 44 users have a member account with submission rights, 9 of which are experts from a validation authority with approval rights. Approximately 4000 RepXML web pages have been visited during the last ten months.

As seen above (Figure 7), a Business Specification may contain an Aggregate Business Information Entity (ABIE), zero or more Basic Business Information Entities (BBIE), Association Business Information Entities (ASBIE) and related documents. Altogether RepXML contains 138 ABIEs, 1500 BBIEs, 213 ASBIEs and 8 documents. The total number of registered RepXML objects is approximately of 2500. The size of the ".sql" backup file of the PosgreSQL backend storage DBMS used by the registry is 7 MB, with 1.9 MB as starting size.

User activity

Regarding the testers there were three main user profiles:

- large companies and associations like Eurofer (European Confederation of Iron and Steel Industries), the BoostAero project (world wide consortium in the air and space industries), and ADAE (Agency for the Development of the Electronic Administration);
- analysts and consultants developing standard Business Specifications on behalf of business communities;
- software editors interfacing their own client application tool with the registry.

At first RepXML was mainly used by users of the first profile that were already familiar with the UN/CEFACT core component approach. However the overall use of RepXML was slow to take off. This was mainly due to the fact that the usefulness of RepXML was not obvious to new potential users. This problem was solved by presenting and demonstrating RepXML on many occasions and by providing some personalized support to users. Subsequently new users not yet familiar with the core component modelling approach found in RepXML an efficient tool for developing, submitting and reusing Business Specifications. In particular the automatic export of stored Business Specifications either in XMI and XML Schema proved extremely useful to them.

Nevertheless, the approval activity has remained slow. It turns out that the research tool implemented in RepXML is not efficient enough for harmonization purposes. This problem appears when a person from the validation authority tries to harmonize and approve submitted Business Specifications. He is faced with hundreds of Business Specifications and consequently has great difficulty to find similarities in business data concepts.

4.6 RepXML conclusion

RepXML provided us with the opportunity to implement a ebXML registry/repository solution to address the storage and publishing of e-business data for companies or administrations. To the best of our knowledge RepXML was the most complete solution based on ebXML standards – namely the ebXML Registry Repository specifications [12][13] and the ebXML Core Component Technical Specification [19] – as it relies on a ebXML registry/repository, a rich interfaces and a remote client tool to provide all the functions necessary to browse, search, create, submit, approve and export Business Specifications. In addition to the actual implementation of RepXML, the field trial that was conducted for several months with real business users provided us with some useful feedback from which we can draw the following conclusions.

At content level, the CCTS standard is probably today, in our opinion, the best approach to develop, manage and reuse generic business data as it provides a very valuable model for defining aggregate business data. However the CCTS approach suffers from the fact that the necessary semantics harmonization between the various core components requires a high level of expertise by the experts from the validation authority. It also suffers from a lack of flexibility and extensibility.

At the registry level, the RepXML project confirmed that a registry conforming to the ebXML Registry Repository specifications is well suited to reference and store UN/CEFACT Core Component-based artefacts. Furthermore we believe that the adaptability and extensibility of the registry information model as defined in [12] allows for other types of e-business artefacts like business processes, collaboration profiles, web services, etc. to be managed by an ebXML registry, although this has actually not been implemented in RepXML.

The RepXML experimentation also showed that there is a need for a reliable registry system that allows users to search for and use business data for their business requirements. As the RepXML field trial was conducted on B2B data displayed both in French and English with users from different countries, we can assume that this is a world wide need. This is actually supported by other similar initiatives that occur in Korea, Canada, etc.

Overall the added value provided by RepXML relies on the fact that it facilitates the creation of new Business Specifications based on existing ones and enables the automatic export of stored Business Specifications either in XMI or XML Schema. This proves that it is possible to hide part of the complexity of the underlying standards and, consequently, that a system like RepXML can become a powerful tool to be used by business analysts at design time. This is even more so when we consider that we did not have the chance to test and implement all services provided in the registry (and defined in [13]) such as federation, content management and notification, simply because it was out of the scope of RepXML. However we believe that a registry application can strongly benefit from these additional registry services and that RepXML could fairly easily be integrated to them.

After analysing our experience in the development and experimentation of RepXML we are confident that we are addressing problems that are relevant to the needs and requirement of many companies and administrations and that we are working in the right direction. This preliminary work on RepXML has emerged onto several open research issues; these are detailed in the next section.

5 Open research issues

In this section, building on our knowledge of the RepXML experience, we detail the open research issues and future work that we will conduct.

5.1 Semantics integration

RepXML has allowed us to identify the limits of a shared data solution that does not have an efficient mechanism to facilitate the matching and the harmonization between existing business data.

If we want a wider use of standard business artefacts, and therefore a wider spread of B2B applications, we must increase the data definition flexibility and introduce mechanisms in solutions able to create business data libraries which are not exclusively based on a controlled vocabulary of terms. In that situation, integration of new concepts requires a tremendous harmonization effort that is too costly in terms of time and realization for SMEs.

Moreover business libraries tend to increase substantially the volume of data, or more generally of semantics concepts, in the professional exchanges area. For instance the UN/CEFACT Core Component draft library released in June 2004 contained 170 components whereas the October 2005 release contains 474 components. Even if the sharp increment is mainly due to the fact that we are in the beginning, it is actually not possible (for the moment) to estimate an upper bound on the number of CCs required. If the B2B is not just a "simple" exchange of orders and invoices, but complex exchanges between partners, the size of the library could become enormous. Without automatic or semi-automatic matching between concepts, it will be practically impossible to make a coherent use of the solution.

Currently the definition of business documents to be exchanged demands a lot of time and knowledge when discovering and reusing business data because the existing solutions are complex. Consequently, this complexity diverts the user from concentrating on his competence when designing the business collaboration. Conversely a system where the definition of business data would be more flexible and tractable by a machine would not require so much effort and knowledge.

Therefore the enrichment of business data with semantics concepts and ontology description of the contents will surely increase the adaptation abilities and construction of new collaborations. Thus the experience conducted in [1], based on the introduction of OWL-L ontologies in ebXML registries, can constitute a good starting point for improving the registry, while maintaining the current RepXML architecture.

5.2 Mediation/distribution

The adoption of an ebXML registry provides several profitable content management services within the e-business applications context. Among these services the registry federation offers the possibility to build a shared storage system distributed on the web for business artefacts, as it is the case for P2P systems. Nevertheless to obtain coherent query results, the registry federation requires shared contents to be homogeneous. Although there is a registry federation proof-of-concept project currently underway in Asia (ebXML Asia Committee), there are no implementation results publicly available yet. Therefore our assumptions are based on theoretical considerations of the specifications [12][13].

This homogeneity requirement may not be a problem in cases like the UN/CEFACT Registry because a standardization body can require a conformity test on the contents' structure and semantics before accepting new registries in its federation. But in the context where business partners are involved in a new partnership, a shared system will in most cases have to deal with an existing heterogeneous base of business artefacts on which to build the business exchanges. Regarding this issue the RepXML architecture naturally leads to the adoption of a mediator that would increase the ability of heterogeneous data systems' integration, as shown in [6].

5.3 Content enhancement

When engaging in business exchanges or new partnerships, the involved parties must have access to a consistent set of business information. In RepXML we have focused on the base components of a business message; even if this improves on the solutions currently available, it is not enough to completely support e-business applications. Also it appears that e-business registries/repositories are starting to acquire a new position with respect to the new emerging service registry and repository for SOA governance [25] [26].

This implies that a solution such as RepXML should be able to manage not only business documents, but also a predefined set of business information necessary to conduct business collaborations. As part of this information set we can list: business processes, service descriptions, company profiles, trading partner agreement, etc. Moreover this information set should be completely defined and universally recognized.

5.4 Design time – Run time Interface

The discovery of existing business artefacts is not limited to the discovery of the meaning of stored business information. It is also very important for users to be able to retrieve the information in various useful machine readable formats. Indeed, the ability to export business artefacts into machine readable formats was strongly welcomed by all participants during the RepXML experimentation.

The problem to solve here is the transformation of part of the output at design time (i.e. the produced documentation or information collection) so that it becomes the primary input at run time (i.e. executable machine code or machine understandable file format). The problem is in the definition of the minimal set of information to be collected at design time and maintained in a repository so as to enable this transformation.

RepXML, which proposes a first solution, has confirmed the feasibility of generating such machine readable formats in an e-business environment. We are convinced that we should further develop this aspect.

6 Conclusion

In this paper we presented the RepXML experience that deals with the problem of building registries for storing e-business artefacts and proposes a solution to problems that currently represent an obstacle for the wider use of B2B mechanisms, in particular the setting up of business collaborations. We mainly investigated a solution for storing business components compliant to the ebXML approach, and provided new interesting material for the improvement of global e-business registries, showing it is possible to build a solution that enables the sharing and reuse of business data.

The RepXML experience also highlighted additional problems and provided new research directions: (i) the current lack of a suitable semantics representation of business data in B2B solutions, (ii) the definition of a coherent standard set of e-business artefacts, (iii) the need of building mediators in heterogeneous data environment and, (iv) the problem of generating automatically a machine interpretable format.

We stress that these four points need to be investigated in order to construct a complete e-business registry solution, and we plan on doing so during our new collaboration between France Telecom and UVSQ.

7 Glossary

ABIE – Aggregate Business Information Entity

ACC – Aggregate Core Component

ASBIE – Association Business Information Entity

ASCC – Association Core Component

ANSI ASC X12 – American National Standards Institute Accredited Standards Committee X12

BBIE – Basic Business Information Entity

BCC – Basic Core Component

BIE – Business Information Entity

CC – Core Component

CCT - Core Component Type

CCTS – Core Component Technical Specification

EDI – Electronic Data Interchange

ebRIM – ebXML Registry Information Model

ebRS – ebXML Registry Services and Protocols

JAXR – Java API for XML Registries

LCM – Life Cycle Manager

OWL – Web Ontology Language

OWL-L – Web Ontology Language Lite

OWL-S – Semantic Markup for Web Services

QM – Query Manager

RDF – Resource Description Framework

SOA – Service Oriented Architecture

SWIFT – Society for Worldwide Interbank Financial Telecommunication

UBL – Universal Business Language

UDDI – Universal Description Discovery and

Integration

UML – Unified Modelling Language

UN/EDIFACT – United Nations Electronic Data Interchange For Administration, Commerce and Transport

UNIFI – UNIversal Financial Industry

XBRL – eXtensible Business Reporting Language

XMI – XML Metadata Interchange

8 References

[1] Asuman Dogac Yildiray Kabak Gokce B. Laleci - Enriching ebXML Registries with OWL Ontologies for Efficient Service Discovery - Proceedings of the 14th International Workshop on Research Issues on Data Engineering: Web Services for E-Commerce and E-Government Applications (RIDE'04)

[2] Hill N. C. and Ferguson D. M. – Electronic Data Interchange: A Definition and Perspective – EDI Forum: The Journal of Electronic Data Interchange: 5 – 12 (1989)

[3] Hofreiter B., Huemer C. and Kim J.-H. – Choreography of ebXML Business Collaborations – UF eGov-Simple project (September 2004)

[4] Hofreiter B., Huemer C. and Klas W. – ebXML: Status, Research Issues, and Obstacles – Proceedings of the 12th International Workshop on Research Issues in Data Engineering: Engineering e-Commerce/ e-Business Systems (RIDE 2002)

[5] Khaled F Abdalla - A model for semantic interoperability using XML - Proceedings of the 2003 Systems and Information Engineering Design Symposium

[6] Tuyet-Tram Dang-Ngoc, Georges Gardarin – Federating heterogeneous data sources with XML

[7] Ballvé D. and Jalonen M. – ebXML Case Study: LomakeFi - Registry-based Environment for Assembling XML Schemas and Electronic Forms for the Finnish Government – lists.ebxml.org/archives/ebxml-mktg-sc/200310/doc00000.doc (September 2003)

[8] France Telecom – RepXML Project – www.disa.org/cefact-groups/icg/downloads/Archives/RepXML_Overview_V1.zip (September 2004)

[9] ISO/IEC 14662:2004 – Information technology - Open-edi Reference Model

[10] ISO 20022 – UNIFI: UNIversal Financial Industry message scheme – http://www.iso20022.org/

[11] Low B. K. and Sloan B. – Current Developments in Internet-based Electronic Data Interchange (EDI) and the Implications for the Construction Industry — ProC-E-Com Working Paper No.3 (December 1999)

[12] OASIS – ebXML Registry Information Model – www.oasis-open.org

[13] OASIS – ebXML Registry Services Specification – www.oasis-open.org

[14] OASIS ebXML Registry TC – Deployment Profile Template For ebXML Registry 3.0 OASIS Specifications – http://www.oasis-open.org/committees/documents.php?wg_abbrev=regrep

[15] OASIS – UBL - Defining a common XML library of business documents – www.oasis-open.org/committees/tc_home.php?wg_abbrev=ubl

[16] REMKO – Korea ebXML Central Registry & Repository – www.remko.or.kr

[17] RosettaNet Business Dictionaries – www.rosettanet.org/

[18] SWIFT – Standards Financial Dictionary – www.swift.com/index.cfm?item_id=2640

[19] UN/CEFACT – Core Components Technical Specification V2.01 – Part 8 of the ebXML Framework (Standard ISO/TS 15000-5:2005) – www.untmg.org/

[20] UN/CEFACT – TBG Library of Core Components and Business Information Entities – www.disa.org/cefact-groups/tbg/wg/tbg17_main.cfm

[21] UN/CEFACT – UN/CEFACT Registry Implementation Specification – http://www.disa.org/cefact-groups/icg/downloads/index.cfm

[22] UN/CEFACT – XML Naming and Design Rules – http://www.disa.org/cefact-groups/atg/downloads/index.cfm

[23] U.S. Office of Management and Budget – Data Reference Model V 2.0 - www.whitehouse.gov/omb/egov/documents/DRM_2_0_Final.pdf

[24] XBRL – How XBRL Works – www.xbrl.org/HowXBRLWorks/

[25] IBM Arms Business Partners to Drive Growth for Mid-Sized Companies Using SOA - http://www-03.ibm.com/press/us/en/pressrelease/19364.

wss

[26] SUN Microsystems – Using an SOA Registry Repository – http://www.sun.com/products/soa/registry/soa_registry_wp.pdf

[27] SUN Microsystems – Java API for XML Registries (JAXR) – http://java.sun.com/xml/downloads/jaxr.html

[28] RepXML Web site – http://www.RepXML.org

[29] freebXML Web site – http://www.freebxml.org