RepXML Project

Version 1.0 – 13 September 2004

Editor(s):
Ivan BEDINI (R&D/BIZZ/PMX) Francis BERTHOMIEU (R&D/MAPS/eCB) Fabrice BOURGE (R&D/BIZZ/PMX) Fabrice JEANNE (R&D/BIZZ/PMX) Sébastien WAFFLART (R&D/MAPS/eCB)

19 pages

Sommaire

Table des figures

1 EXECUTIVE OVERVIEW

1.1 Introduction

The ebXML framework defines a set of rules that business communities or professional sectors can use to register their data and their business practices. Then companies or administrations can use these business data or practices through the ebXML standard infrastructure, which is primarily based on 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. A lot of work still needs to be done to reach this maturity and to favour a significant growth in the ebXML use by companies and administrations.

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 sector. Some of the standardisation work currently done at UN/CEFACT and OASIS aims at filling this gap much needed by companies and administrations.

1.2 Business needs

Whether they are professionals, software developers, information system designers, or simple users in SMEs, in large companies, or in administrations, many people need to have easy access to a common data base where they can find answers to their questions pertaining to their business activity, for example:

- What models can I use to exchange documents with my business partners?
- What are the usual practices in my business sector?
- I must ensure compatibility with model XYZ; what's in this model? How is it organised? What's the meaning of all objects referenced in it?
- What normative initiatives have been taking place in my environment? What is the methodology used in my domain?

Public sharing of this type of information shall contribute to the open exchange between current or future business partners. With such a data base everyone is able to adapt faster and more easily to its business environment. The result is a better consistency and harmonisation of methods and practices within a given business sector.

1.3 Project objectives

The main goal of the RepXML project is to provide a complete set of francophone business components that is accessible on the Internet. Eventually these components will also be provided in other languages, while rendering them understandable by francophone users.

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

- Define and implement the procedure that enables business sectors to submit, retrieve and re-use e-business components (francophone BIEs);
- Implement a Registry/Repository that registers and stores these components;
- Develop a "Connector" that allows business sectors to create off-line new BIEs;
- Develop the interface between the Connector and the Reg/Rep server;
- Conduct a field trial with EDIFRANCE members, starting in October 2004.

2 PARTNERS

2.1 EDIFRANCE

Created in 1990 EDIFRANCE is a not-for-profit organisation that promotes and develops ICT (Information Communication Technology) among French companies and administrations.

With 150 members and thousands of indirect members, EDIFRANCE gathers together private companies, administrations, business communities, professional organisations, as well as software and service providers for Paperless Trade & e-Administration applications.

EDIFRANCE's missions are:

- Acknowledging and understanding the ICT's stakes;
- Sharing, cooperating, informing, training, guiding, promoting with respect to ICT.

EDIFRANCE submits contributions to standardisation groups such as UN/CEFACT, W3C, and OASIS.

EDIFRANCE is the RepXML project owner.

2.2 France Telecom

France Telecom is a well known global telecommunications operator present in 220 countries and areas across the world.

France Telecom offers a wide range of Corporate Solutions, especially in the domains of EDI and VAN (Value Added Network). For the past few years, the R&D Division has been studying closely the ebXML framework through several internal projects.

France Telecom is co-project manager for RepXML, responsible for the Registry/Repository module and access to it.

2.3 SRCI

SRCI is a small-sized software company specialised in EDI and process oriented interoperability. Its domains of expertise are EDIFACT and ebXML, both in BtoB and BtoA environments, and its business ranges from auditing, to consulting, to project managing, and to final implementation.

SRCI is an active EDIFRANCE member.

SRCI is co-project manager for RepXML, responsible for the Connector module.

3 ENVIRONMENT

3.1 Specifications

Several specifications are being considered during the RepXML project. These specifications are listed below:

CCTS 2.01: The Core Components Technical Specification (CCTS – ISO 15000-8) constitutes Part 8 of the ebXML Framework. This specification defines building blocks, either context neutral (Core Components) or context specific (Business Information Entities), that are stored in the ebXML Registry.

UML 1.5: The Unified Modeling Language (UML) is a specification defining a graphical language for visualizing, specifying, constructing, and documenting the artifacts of distributed object systems. UML being the essential component of the UN/Cefact methodology (UMM), all library components (CCs and BIEs) shall be expressed in UML.

ebXML Registry 2.5: The ebXML Registry Services (ebRS – ISO 15000-4) provides a set of services that enable sharing of information between interested parties for the purpose of enabling business process integration between such parties based on the ebXML specifications. The shared information is maintained as objects in a repository and managed by the ebXML Registry Services.

The ebXML Registry Information Model (ebRIM - ISO 15000-3) provides information on the type of metadata that are stored in an ebXML Registry as well as the relationships among metadata classes. The RIM defines in a generic manner what types of objects are stored and defines how stored objects are organized, but it does not specifically address the storage of CCs and BIEs.

A syntax neutral model and a format based on which must be defined allowing CCs and BIEs to be bound to the Registry.

3.2 freebXML

freebXML.org is an initiative that promotes open source development of ebXML-based applications. Among the different freebXML projects, *ebXML Registry/Repository* is a general registry adopting a generic and extensible information model that includes the ability to have arbitrary associations between entries in the registry. The repository is a way of storing information about entries in the registry. Any type of data can be stored in the repository.

The *ebXML Registry/Repository* is used to reference and store common pieces of business data thus allowing users to compose B2B messages.

3.3 LomakeFi initiative

Republica's LomakeFi Form Assembler tool is one of the results of a project that aimed to produce electronic forms for the Finnish Government based on existing paper forms. It uses the ebXML Core Component approach to define the form parts and relies on ebXML Registry to store information about these parts. This case study presents this tool, the registry-based environment behind it and the work developed by Republica Ltd for this project.

The LomakeFi initiative was a good source of inspiration to the RepXML project.

4 PROJECT DESCRIPTION

4.1 General principles

RepXML is a Registry/Repository where business components are registered and stored. These business components can be composed of BIEs, XML Schemas and/or business documents. In the remaining part of the document, these business components are called *Business Specifications (BS)*.

The main purpose of RepXML is to allow users to prepare and submit to the server new proposed *BSs* for validation, and to retrieve validated *BSs* from the server.

RepXML can be accessed via a specific software application (called the Connector) or via a Web interface. Using the connector a user can prepare off-line a new proposed *BS* using some sort of work sheet, and then submit this work sheet to the server.

For clarification, it is convenient to distinguish between several categories of users interacting with RepXML:

- Author: a person that creates a new *BS* (eg for the needs of a particular business sector);
- Submitter: a person or an organisation that submits a newly created *BS*;
- Validator: a person or a group of persons from a validation authority that validates the new proposed business components (eg UN/CEFACT)
- Guest user: a person from a company or an organisation that retrieves validated business components for their specific needs;
- Administrator: a person from the organisation in charge of managing and maintaining the server.

4.2 Architecture / modules

The overall functional architecture is shown in Figure 1. The two main modules are the Connector and the RepXML application.

4.2.1 Connector

The connector is a piece of software that allows submitting organisations to create *BSs* and to submit these to a validation authority via the RepXML application. Before creating a *BS*, one has the possibility to perform a semantic search on the valid BIEs, either locally in the data base attached to the Connector or in the registry of the remote RepXML application.

When submitting a *BS* to the RepXML registry, a specific submitting procedure is started which performs some consistency control before the actual sending out.

BS forms can be downloaded from the RepXML Registry to the connector on demand or in a systematic manner so as to update the *BS* forms that are locally present in the connector.

The connector may also be used by guest users to retrieve from the RepXML Registry pieces of information in different formats: either existing documents (eg PDF, EDIFACT, ASN1…), or automatically generated documents (UML models of BIEs in PDF or XMI formats, or XML Schemas of BIEs).

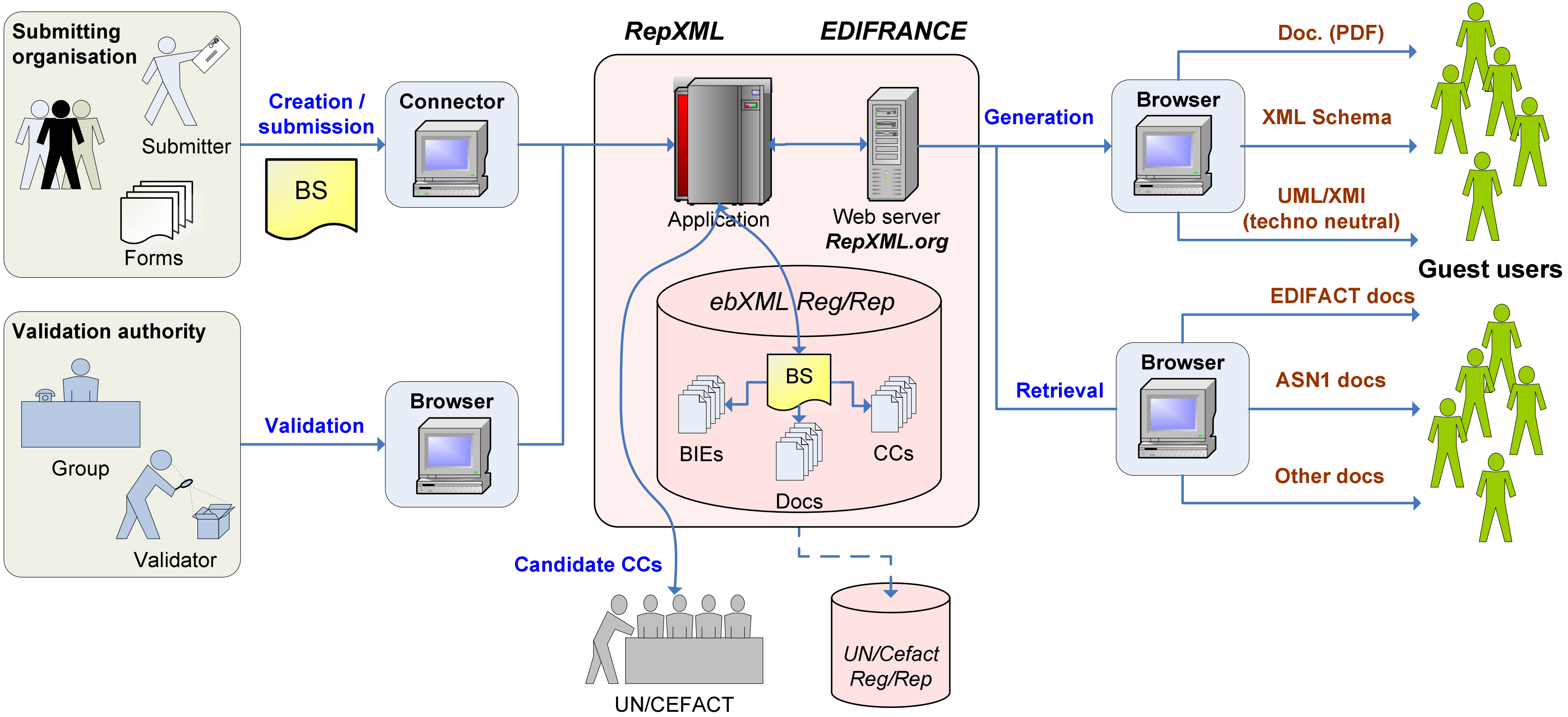

Figure 1. Functional architecture

4.2.2 RepXML application

The RepXML application is based on existing components like the ebXML Registry/Repository (freebXML) and a key registry (keystore). These components are interfaced with the RepXML application without being modified. The ebXML Registry/Repository provides a set of methods that are accessible to the RepXML application for storing, searching and organising objects.

A Web Server (RepXML.org) is interfaced with the RepXML application. It allows guests users (development teams, IS designers…) to access and visit the Web site, and to retrieve RepXML *BSs* in different formats.

4.2.3 RepXML Business Specification

A RepXML *BS* is the representation of a business requirement that a business sector wants to submit. After validation the *BS* will be stored in the RepXML Registry/Repository.

On the Connector a specific *BS* is created using a form. Between the Connector and the RepXML Application a *BS* is submitted as an XML Schema.

A RepXML *BS* is an information set that enables to qualify one or more ABIEs with its BBIEs, ASBIEs, DTs and attached documents (XML Schema, legal documentation, user documentation, UML description…). Figure 2 below shows the structure of a *BS*.

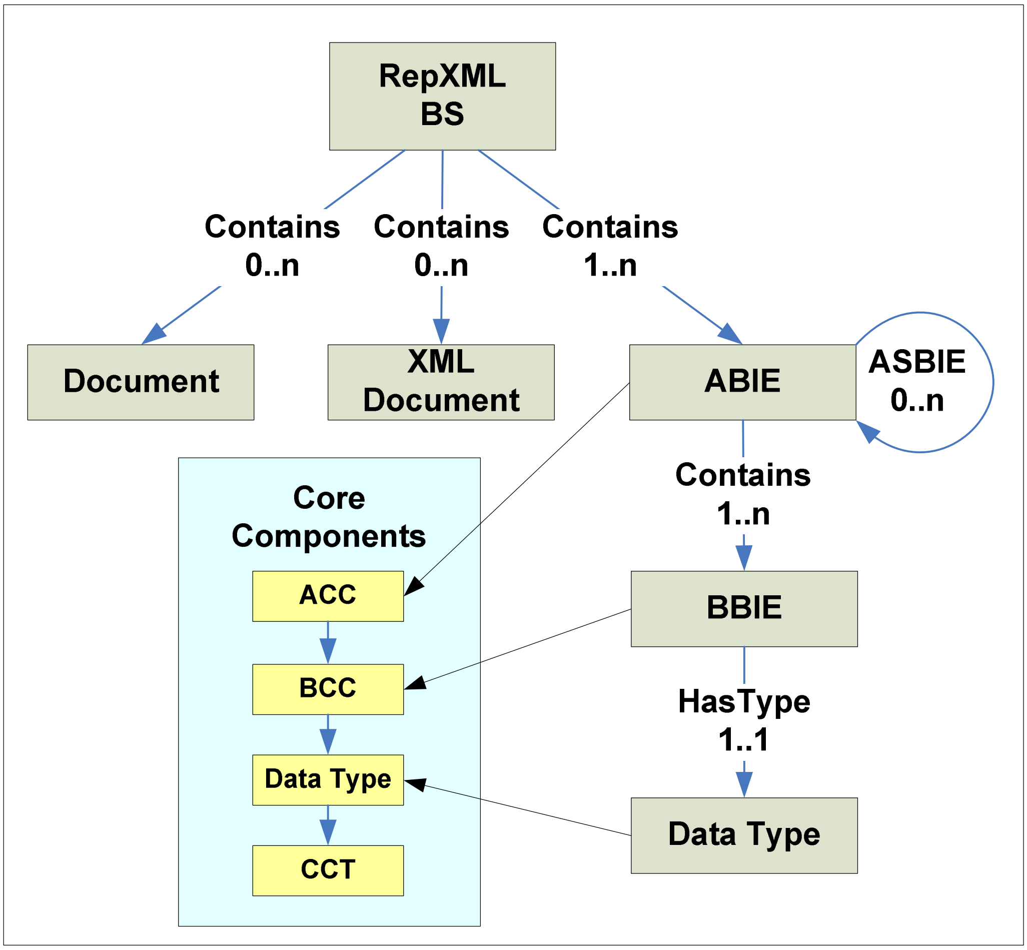

Figure 2. RepXML BS definition

4.3 Submission / validation procedures

4.3.1 Submission procedure

This procedure describes the sequence of actions/exchanges between the submitting organisation and the RepXML application. There are three steps in this procedure:

1. Synchronization of data between Connector and RepXML
2. The author creates a new *BS*
3. The *BS* is submitted to RepXML.

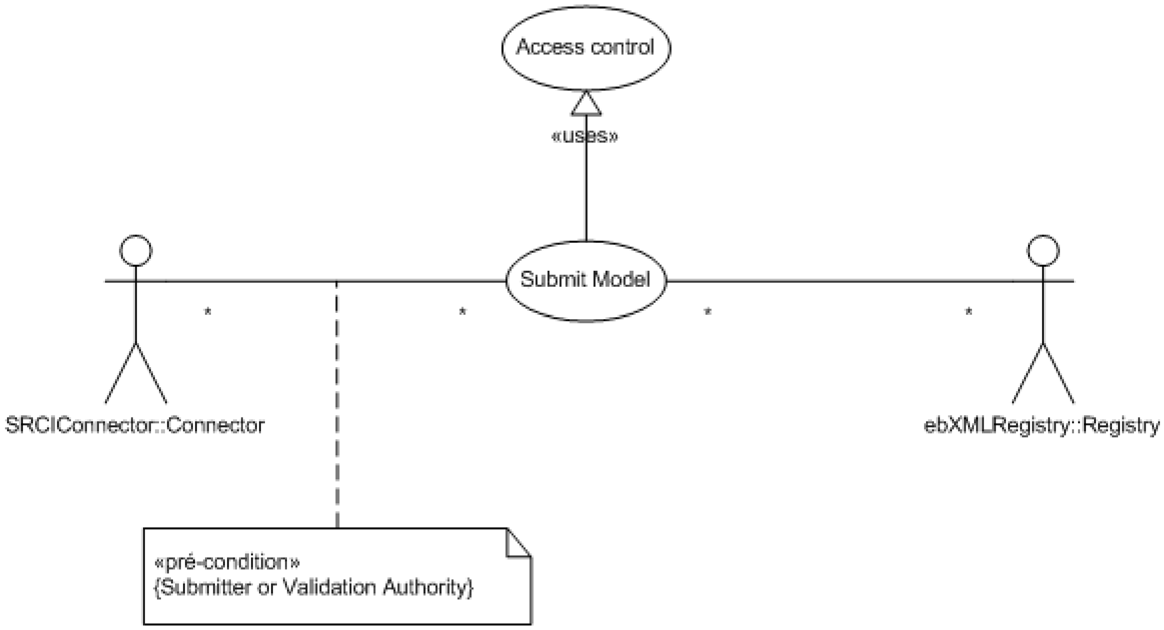

Figure 3. "BS submission" use case

4.3.2 Validation procedure

This procedure describes the sequence of actions/exchanges between the validation authority and the RepXML application. There are three steps in this procedure:

4. The Validator is informed that there has been a new *BS* submitted
5. The Validator evaluates the new *BS*
6. The Validator validates or rejects (with justifications) the submitted *BS*.

In the case where the Validation Authority thinks that the submitted *BS* requires the creation of new Core Components, then there is an additional step in this procedure:

7. Candidate CCs are proposed by the Validator to the UN/CEFACT.

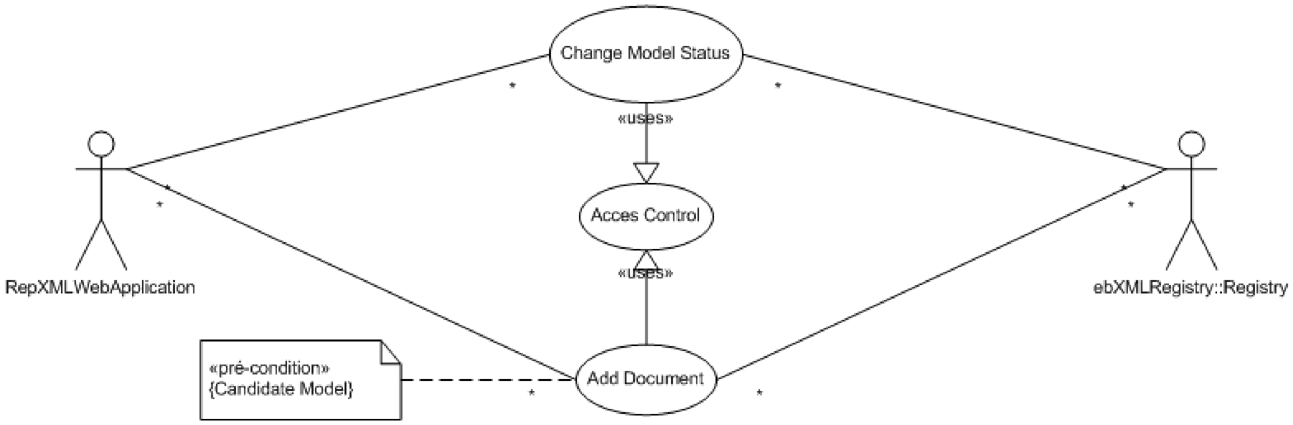

Figure 4. "BS validation" use case

4.4 Storing CCs/BIEs in RepXML

Regarding the storage of CCs and BIEs in the Registry, we have used the same metadata approach as Republica in the LomakeFi registry initiative, that is CCs and BIEs are stored as registry entries. However we also have the possibility to store items such as documentation (eg in PDF) in the Repository.

Storing CCs and BIEs follows a specific *BS* based on the ebXML Registry Information Model (ebRIM). The details of this model which is very to the one used by Republica are as follows:

- A BS is stored as an ExtrinsicObject of type “Model”
- An ACC is stored as an ExtrinsicObject of type “AggregateCoreComponent”
- A BCC is stored as an ExtrinsicObject of type “BasicCoreComponent”
- A CCT is stored as a ExtrinsicObject of type “CoreComponentType”
- An ASCC is stored as an Association of type "AssociationCoreComponent"
- An ABIE is stored as a ExtrinsicObject of type “AggregateBusinessInformationEntity”
- A BBIE is stored as an ExtrinsicObject of type “BasicBusinessInformationEntity”
- A BC is stored as an ExtrinsicObject of type “BusinessContext”
- An ASBIE is stored as an Association of type "AssociationBusinessInformationEntity"
- A DT is stored as an ExtrinsicObject of type "DataType"
- ACCs/ABIEs are associated to BCCs/BBIEs with “Contains”, respectively
- BBIEs are associated to DTs with “HasType”
- DTs are associated with CCTs with "Refine"
- CCTs are associated with Content Components and Supplementary Components with "Contains"
- BIEs are associated to CCs with “Refine”
- A BC is associated to BS with "ContextFor"
- A BC is composed of 8 context categories; Context Categories are stored as ClassificationSchemas
- Contexts are mapped to ClassificationNode
- DictionaryEntryName and Definition are mapped to RegistryObject’s name and description, while all the other fields are mapped to Slots.

We have developed an API for this *BS*, which runs on top of JAXR (Java API for XML Registries).

For the time being a BIE only points to the identifier and name of its corresponding CC. This is a pragmatic approach that allows us to store BIEs without having to wait for CCs to be standardised. When CCs are available, then they will be stored in the registry and storing of BIEs will be adapted if need be.

Figure 5 below provides an example of how the Person *BS* is stored in the registry.

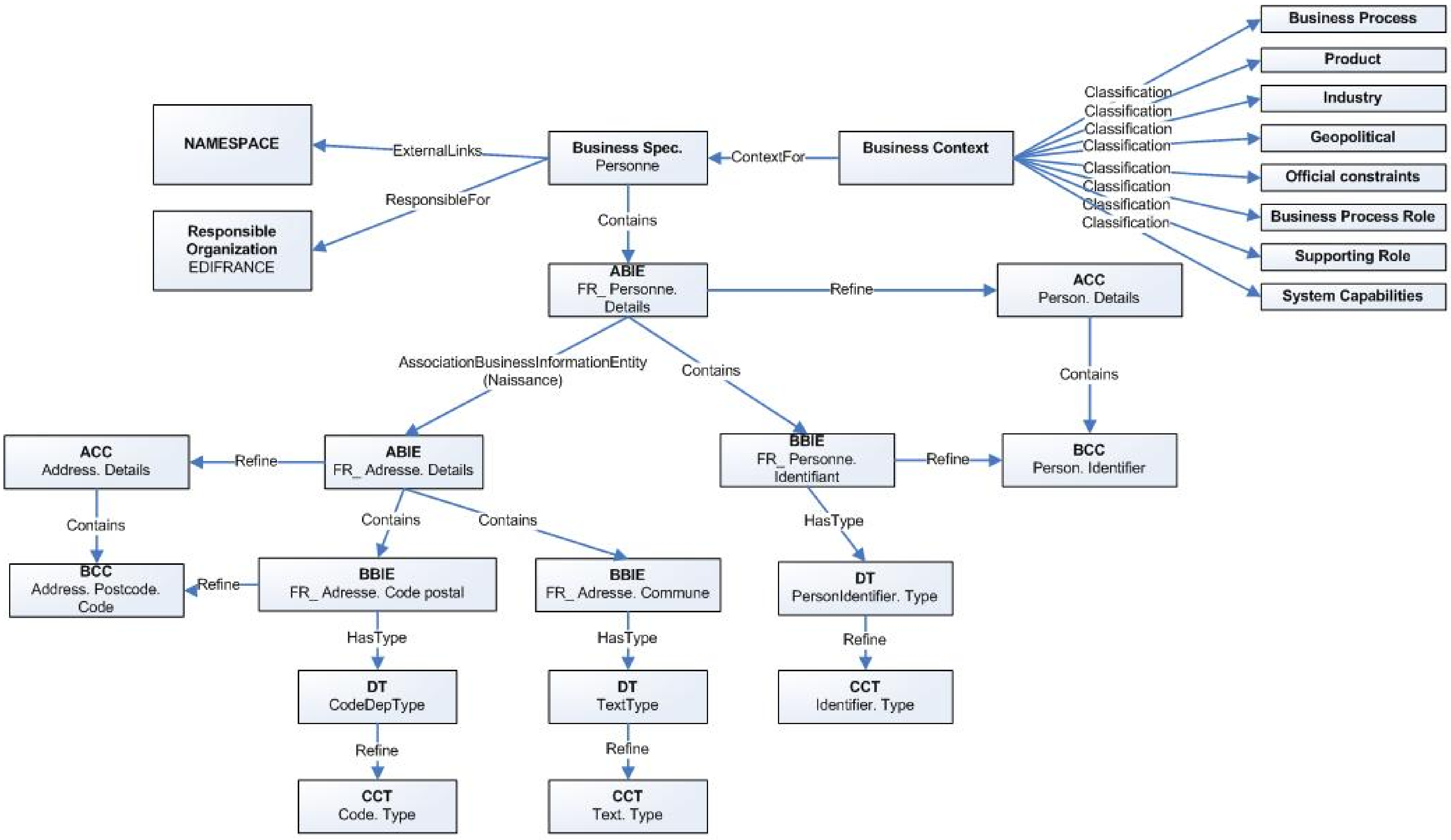

Figure 5. Storing Person BS in the Registry

4.5 CCTS compliance

The four following figures illustrate the level of compliance with CCTS. Each figure has been extracted from Section 7 of CCTS.

The first main difference between CCTS and RepXML is that the Class "RepXML BS" has been added in the RepXML application. One can see the implication of this addition in Figure 9 hereafter (Registry Metadata).

Secondly, in the RepXML application, the property "Unique Identifier; 1..1" has been deleted and replaced by a multi-valued link to the class "UID". This extension enables to fit with the business needs where a component can be identified with several IDs, each depending of a specific organisation.

Thirdly, in the current RepXML application, the versioning is managed at the *BS* level and not at the level each element. It seemed more efficient to manage the versioning in this way because a RepXML *BS* represents a business-understandable self-sufficient piece of information.

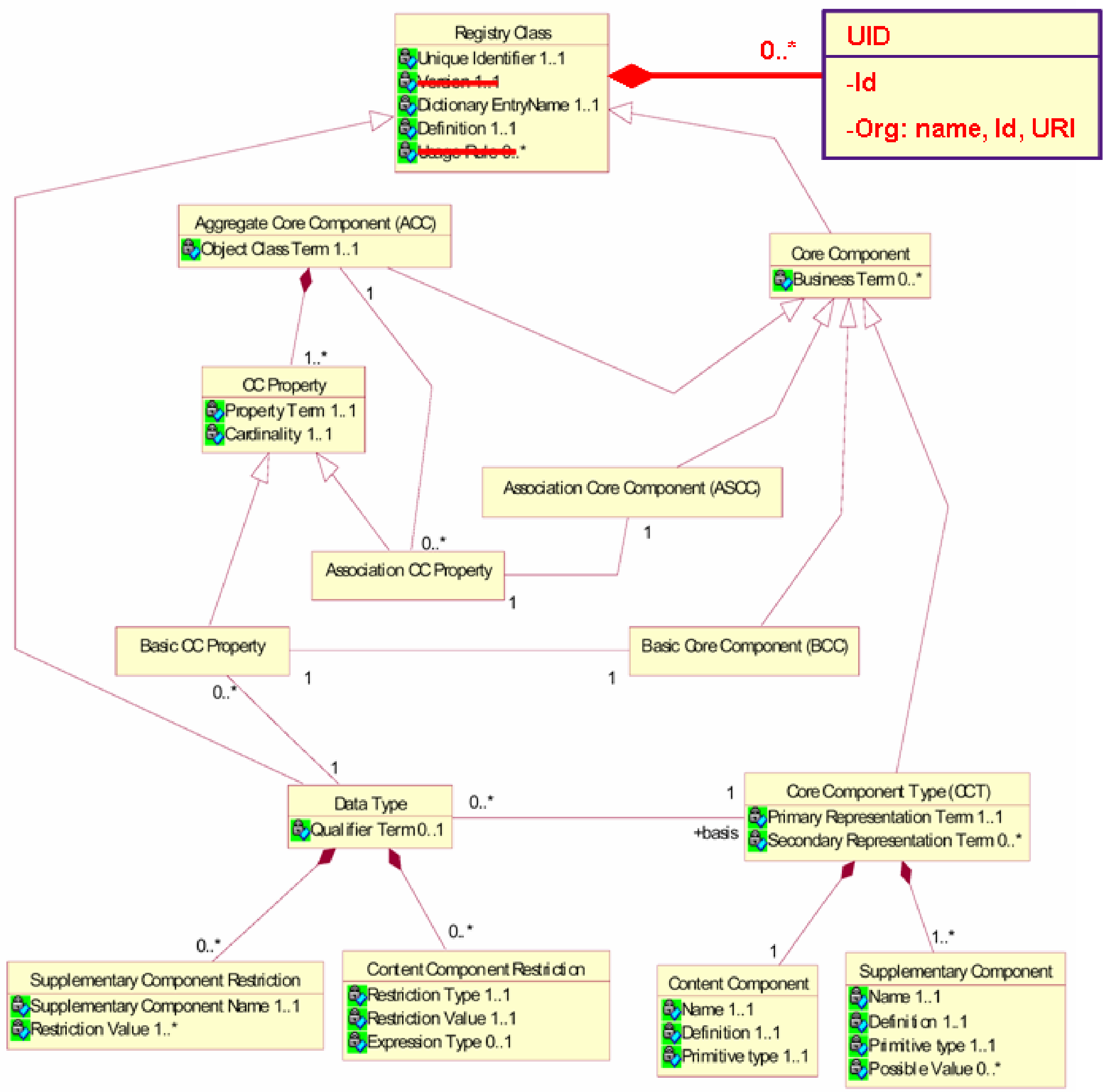

Figure 6. Core Components and Data Types – Full definition

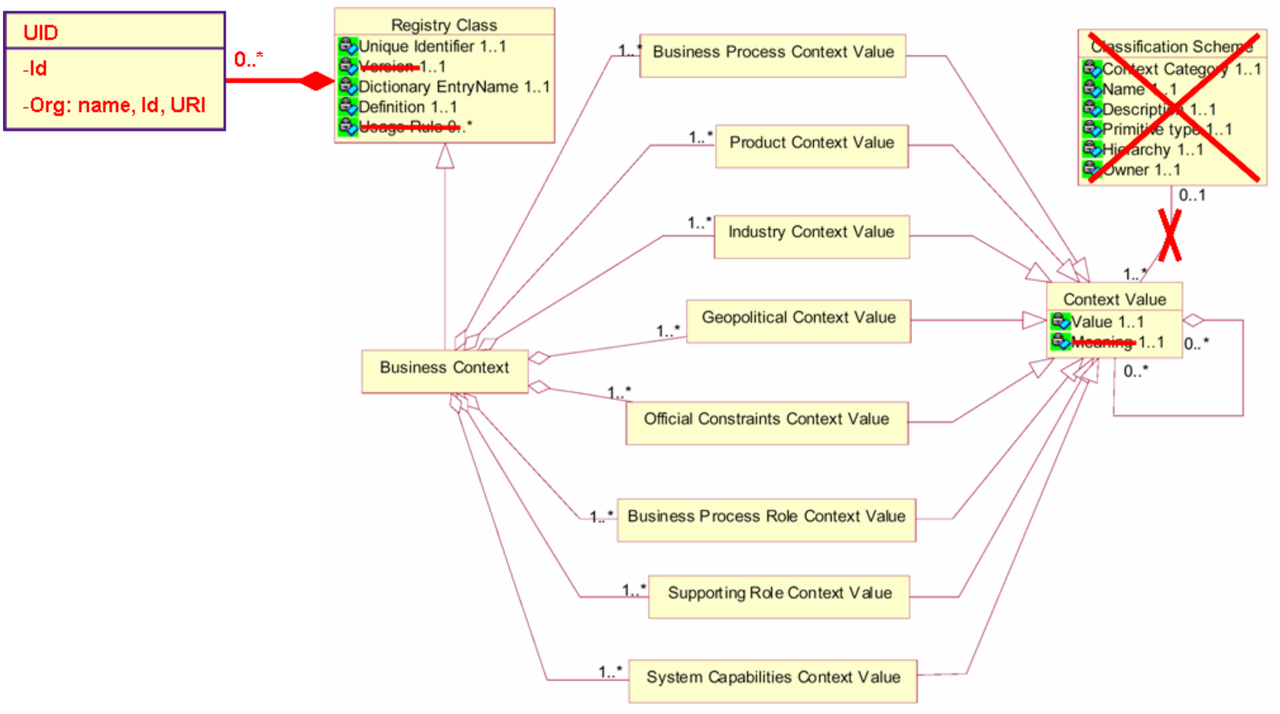

Figure 7. Core Components context definition model

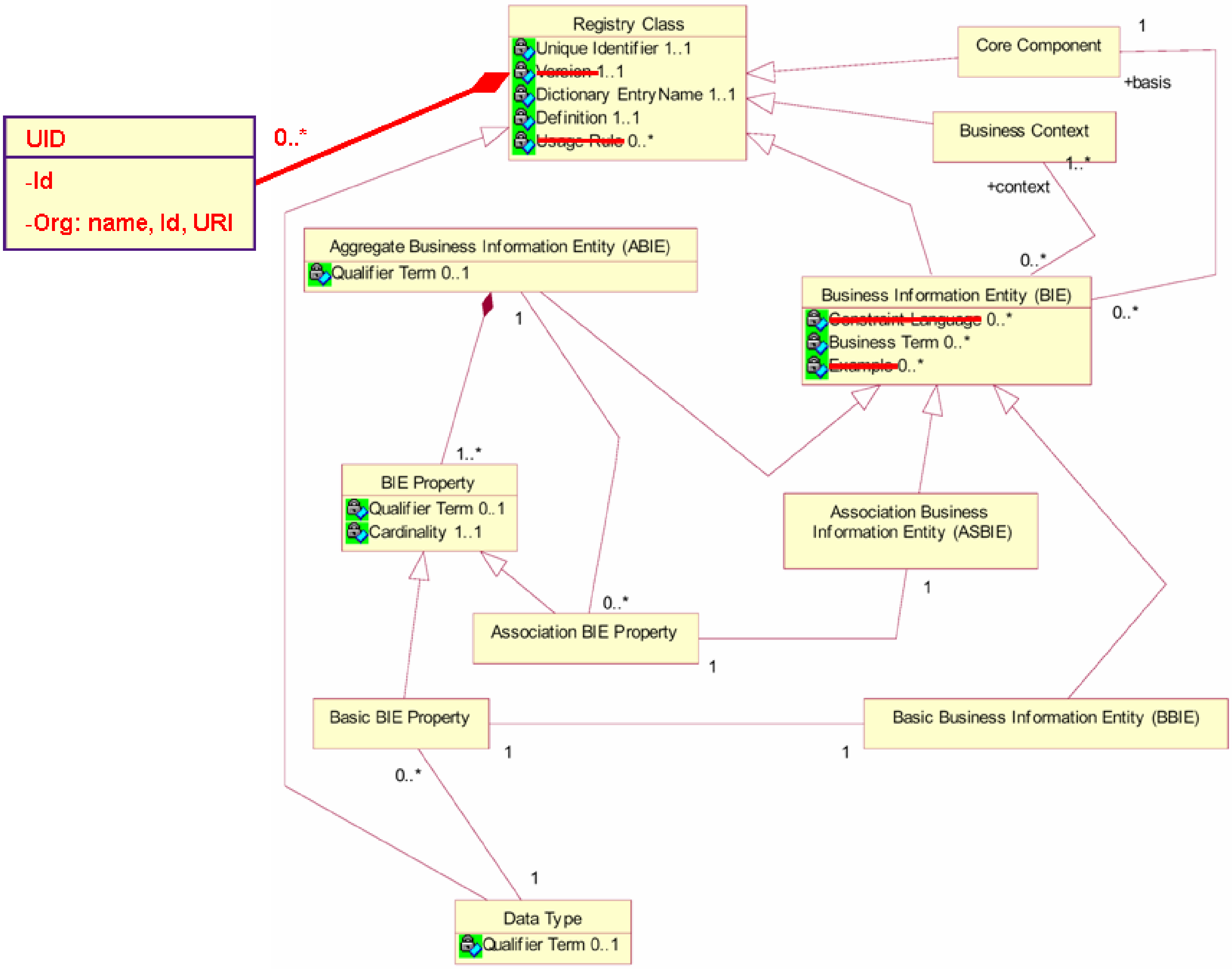

Figure 8. Business Information Entities – Full definition

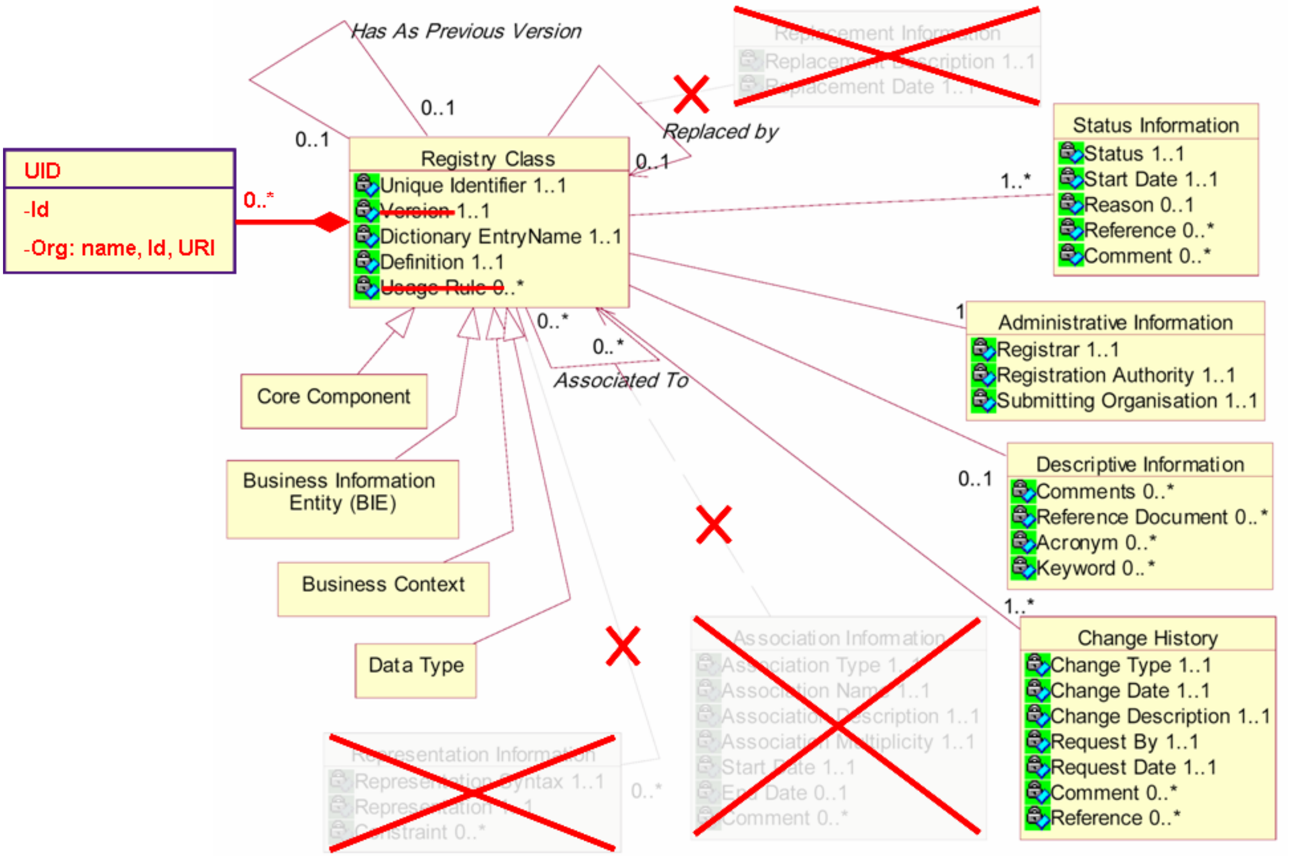

Figure 9. Registry Metadata

4.6 Field trial

A field trial will be conducted amongst EDIFRANCE's members. The trial will help to evaluate:

- the acceptability of the service by end users;
- the reliability of the system.

A detailed report will be established at the end of the project.

5 CONCLUSION AND FUTURE PLANS

RepXML project is a first step towards the deployment of a Registry/Repository for francophone e-business components.

This first step will show the acceptability of the service by the end users (business sectors and standardisation members) as well as the reliability of the platform. The approach for a global deployment is based on "application iterations". In this way the current version enables to create, store and retrieve only ABIEs and not more complex components such as messages or business processes.

In addition, the current version does focus on technology neutral elements and not on a syntax-binding process: a few automated transformations have been implemented (excel sheets to ebRIM, ebRIM to XMI, ebRIM to XMI, and ebRIM to PDF for user documentation).

The future application iterations will take into account the management of "Packages" and business useful automated transformations such as ASN1, XHTML and EDIFACT.

Due to a lack of existing public business Registry/Repository implementations, RepXML could be seen as an isolated application. However many business sectors have already prepared BIEs and Business Processes and are waiting for Registry/Repository applications to appear. Hopefully the UN/Cefact, where the Core Components should be defined and stored, will provide a Reg/Rep in a near future. In the same way, hopefully, francophone sectors (eg the French administration, industry,…) will provide sectorial Reg/Rep applications next year.

With respect to the standardisation, RepXML is close to being CCTS compliant and is based on the Open Source FreebXML application that complies with the ebXML Registry/Repository specifications. The standardisation compliance will lead to further more global applications based on the new e-Business specifications (paradigms).

The main reason why RepXML is not fully CCTS compliant is because the current CCTS has never been developed thoroughly and several inconsistencies and missing parts were identified during the project.

The RepXML Registry/Repository is based on FreebXML V2 that implements a subset of ebXML RR Specification. During the RepXML development phase FreebXML V3 was not reliable enough to be used.

RepXML represents a valuable input for the domestic and international standardisation process. Several RepXML team members are involved in the standardisation process at OASIS RR TC, UN/Cefact and EDIFRANCE.

6 APPENDIX A – GLOSSARY

ABIE

Aggregate Business Information Entity

ACC

Aggregate Core Component

ASBIE

Association Business Information Entity

ASCC

Association Core Component

ASN1

Abstract Syntax Notation One

BBIE

Basic Business Information Entity

BC

Business Context

BCC

Basic Core Component

BIE

Business Information Entity

BS

Business Specification. A RepXML *BS* is the representation of a business requirement that a business sector wants to submit.

CC

Core Component

CCT

Core Component Type

CCTS

Core Components Technical Specification

DT

Data Type

ebRIM

ebXML registry information model

EDIFACT

Electronic Data Interchange For Administration, Commerce and Transport

ICT

Information Communication Technology

JAXR

Java API for XML Registries

RepXML

Registry Repository using XML format

UML

Unified Modeling Language

XHTML

Extensible HyperText Markup Language