

# ECONSTRUCT: EXPECTATIONS, SOLUTIONS AND RESULTS

SUBMITTED: July 2001

REVISED: November 2001

PUBLISHED: December 2001 at <http://www.itcon.org/2001/13/>

*Frits Tolman, Professor  
TU-Delft, The Netherlands  
email: f.tolman@ct.tudelft.nl*

*Michel Böhms, Dr.  
TNO Building Construction Research, The Netherlands  
email: M.Bohms@bouw.tno.nl*

*Celson Lima, Dr.  
CSTB, France  
email : Celson.Lima@cstb.fr*

*Reinout van Rees, Mr.  
STABU, The Netherlands  
email: R.vanRees@ct.tudelft.nl*

*Joost Fleuren, Mr.  
TU-Delft, The Netherlands  
email: J.Fleuren@ct.tudelft.nl*

*Jeff Stephens, Mr.  
Taylor Woodrow Construction, United Kingdom  
email: jeff.stephens@uk.taylorwoodrow.com*

*SUMMARY: The IST 10303 “eConstruct” project (eCommerce/eBusiness in the Building and Construction industry: preparing for the new Internet) is now (November 2001) in its second and final year. The objective of eConstruct is to develop, evaluate and demonstrate how the next generation Internet can be used to improve meaningful communications in the European Building and Construction industry, supporting future eCommerce and eBusiness. The paper presents the goals of the project, discusses the chosen solution, evaluates the results obtained so far and recommends directions for further R&D. Though the paper globally presents the results of the project as a whole, typical eCommerce aspects will be dealt with in more detail (other papers describing other eConstruct results in detail have already been published, or will be published shortly).*

*KEYWORDS: electronic communication, PDT, Next Generation Internet, XML, SOAP, eCommerce/eBusiness, Building and Construction.*

## 1. BACKGROUND

The European BC industry is suffering from its fragmented and multi cultural organisation. Especially in large-scale one-of-a-kind construction, costs of failure form a substantial part of the total project costs, certainly if costs of failure is seen in its broadest meaning - exceeding the common failure costs due to construction errors, time delays and requirement changes - to include those cost elements that are outside the responsibility of the project partners, i.e. inefficiency due to communication problems, or outside the scope of the project, i.e. costs by others due to traffic jams, health problems and accidents, environmental pollution, juridical hassling, and client dissatisfaction.

Until recently there was no great incentive for the industry to change. Clients reluctantly paid the 'additional' costs and if necessary the juridical costs, and competitors were more or less in the same league. Slowly, but surely, the BC industry is changing with many clients no longer accepting that they are not in control, and not accepting costs of failure (both in the process and in the resulting artefacts). Increasing pressure to shorten lead times, and deliver value for money is the general result. In addition, society and its governmental representatives increase their demands on safety, environment, energy and such, resulting in even more regulations, more red tape and more information to process and communicate.

The Building and Construction industry is trying to respond to the changing requirements. Several ways to increase its competitiveness are being developed. Organisational approaches result in new contract types like Design-Build-Operate. Car manufacturer's experiences in Lean Production result in Lean Construction. In addition, the application of Information and Communication Technology (ICT) has stimulated several new approaches. Computers and computer networks are masters in information processing and communication. Costs of failure somehow always result from *miscommunication*. No wonder that ICT is seen as one of the cornerstones of the solution.

Currently the use of ICT is very limited. Every professional uses computers, but no electronic semantical communication is possible and ICT is not used on the project level. Currently communicating in and over large-scale construction projects is still done by traditional means: drawings, fax, mobiles and meetings.

One of the negative results of the current ICT use is that humans function in the information streams as translation or transformation machines. They receive information on paper, extract the input required for their applications, perform their calculations, extract the required output and put that information again on paper to pass it down to others. This is, in terms of Lean Construction, a transformation process without added value (on the contrary it is error prone) that is seen as waste and should be eliminated.

Eliminating humans from the information stream requires an Internet-based Communication Technology (CT) that is applicable for meaningful electronic communication (1) between humans, (2) between humans and computer applications, and (3) between computer applications. Only if humans (parties involved in a project) and computers share a common ontology of construction terms and definitions, then electronic communication becomes feasible. An ontology is a domain specific set of notions, while a taxonomy is an implementation of an ontology.

The quest for electronically available construction semantics shared by humans and computers is not new. Research and Development in this area started already two decades ago with Product Data Technology work in ISO-STEP and IAI. Unfortunately, for Building and Construction this work has not yet produced the required results. The main problem is that this approach requires:

- the development of a CT that serves the purpose;
- international standardisation of the CT;
- wide support of the standard by the software application vendor community, which
- first has to sell many systems, so that
- the decision to use the CT in a project involving many parties can be made
- after which another set of problems will have to be solved (acceptance, errors, incompatibility);

Not really a wonder that the uptake of PDT in BC takes so long.

With the introduction of the new XML-based Internet another ICT approach becomes feasible. XML supports a much more flexible, even anarchistic, approach to meaningful electronic communication. This new opportunity triggered the eConstruct partners (see appendix A) to form a consortium to develop and submit a proposal, see chapter 2.

Though eConstruct was among the first projects looking into this new technology in the Building and Construction industry, it is by no means the only one. Among several others Bentley's aecXML initiative and IAI's (International Alliance for Interoperability) ifcXML initiative are perusing related and overlapping goals,

in the case of the IAI-IFC in close cooperation with eConstruct (as further explained in section 5). Recently XML is becoming a real hype and many related developments have been started up; too much to emphasis in this paper.

## 2. ECONSTRICT PROPOSAL

The next generation XML-based Internet provides a new opportunity for the Building and Construction industry to improve its ability to communicate project information over the Internet. Unlike earlier projects like ATLAS (Ep 7280), CONCUR (Be 96-3016) and others that mainly focused on meaningful communication of design information, eConstruct focuses on eCommerce/eBusiness for a more powerful drive.

The aim of the project can best be illustrated by quoting the eConstruct Technical Annex:

*The 'eConstruct' project aims to develop, implement, demonstrate and disseminate a new Communication Technology (CT) for the European Building and Construction industry, tentatively called Building and Construction (BC) extensible Mark-up Language (bcXML). This Communication Technology will provide the European BC industry with a powerful and low cost communication infrastructure that:*

- Supports eCommerce/eBusiness between users and suppliers of building materials, components, systems and services.
- Is integrated with eCommerce and Design/Engineering applications, and
- Supports virtual market places over the borders of the individual European member states.

Besides the Communication Technology, eConstruct also aims to develop and demonstrate a number of applications that use the CT.

The following applications of bcXML will be developed and demonstrated in the project:

- bcXML support for supply-side information providing, i.e. catalogue building.
- bcXML support for shopping and buying of individual components.
- bcXML support for project related procurement involving design/engineering.
- bcXML support for Computer Aided Selling.

Support for publishing supply-side information using the current Internet is currently offered by a proliferation of initiatives (portals, virtual marketplaces, and such), all using their own methods, formats and rules. Support for shopping and buying individual components over various sources of supply-side information is non-existent. Support for eProcurement directly and openly linked to a product model is non-existent. Computer Aided Selling systems that can automatically input supply-side information from different sources are also not yet available.

## 3. TECHNOLOGY

The aim of eConstruct is inspired by the latest developments in Web technology. The next generation XML-based Internet will remedy the problems with the current HTML based Internet, it will be safe, fast and much more structured. The basic principle is that XML makes a distinction between "content" and "mark-up". The content can be made very BC specific, i.e. the computer 'knows' what we mean by Column, Beam, Floor plan, Height, Weight and such. By splitting content and mark-up it becomes possible to concurrently use different mark-ups *for the same content*, making it for example possible to communicate about a floor slab with a number of openings and produce: human readable descriptions of the slab in different native languages (even with different alphabets) rendered in HTML or PDF, computer formatted descriptions that directly become input in application systems, 2D images of the slab using SVG (Scalable Vector Graphics) (SVG, 2001), and 3D Virtual Reality images using X3D (the XML version of VRML) (X3D, 2001).

Besides the division between content and mark-up, XML is also quite capable to handle distributed data (by using URLs and such), it can easily transform and filter information on-the-fly by using XSLT (eXtensible Style sheet Language Transformation) (XSLT, 2001) and last but not least it allows its users to build XML vocabularies with domain specific definitions of objects and properties; this last XML capability is used by the

consortium to build a vocabulary with BC domain specific semantics. In (Zapthink, 2001) the pros and cons of XML are most clearly evaluated.

#### 4. EXPECTATIONS

The eConstruct partners wanted to develop a Communication Technology (CT) that can play a central role in the construction process, but (at least initially) with a main focus on eCommerce type of applications. In the past most efforts targeted, with limited success, communication in the design/engineering stage. Meaningful communication in design/engineering in construction requires international agreements (standards like ISO-STEP) about a tremendous amount of highly specialised information, which is very hard to achieve. With eConstruct the partners aimed to involve participants active in the Supply Chain, but with a bridge towards design/engineering, project planning and realisation life-cycle stages.

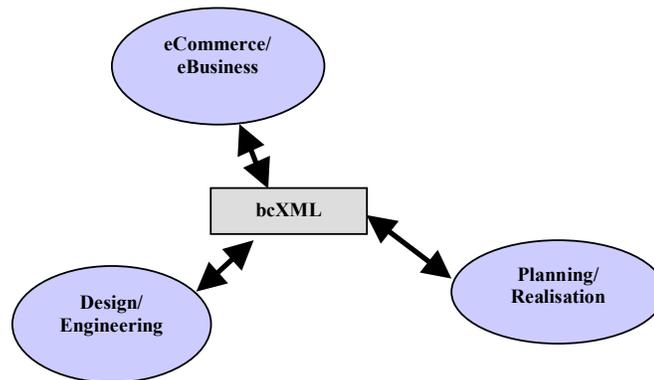


FIG. 1: Focus of bcXML

The partners expected to be able to develop the required Internet-based Communication Technology, together with a number of applications, and to demonstrate electronic meaningful communication over the Internet, over the national borders (in various native languages), with graphical support, and without the pitfall of international standardisation and vendor implementation. By providing a cheap but powerful Internet technology and a mechanism to communicate over different taxonomies, countries and disciplines the partners hope that gradually BC's paper-based Information System will be replaced by an electronic Information System.

#### 5. SOLUTION CONCEPT

Developing a new CT in the rapidly changing XML arena is like shooting on a running target while sitting on the back of a horse. At the start of the project on January 2000, XML usage was limited by the DTD meta-schema. As this meta-schema does not support a number of language constructs required for BC (mainly related to the definition and handling of properties and units), eConstruct started with the development of the bcXML meta-schema and – in co-operation with the IAI-IFC – with the Common Object Schema (COS).

The resulting meta-schema provides interoperability between eCommerce type of applications and design/engineering type of applications.

About one year later however, i.e. January 2001, the XML community started to use XML Schema, the DTD successor. XML Schema supports more language constructs required by BC. With XML Schema it becomes possible to develop a bcXML version that is much more readable and XML spirited. The overall architecture, shown below in Figure 2, shows how applications in various sectors are provided with meaningful and rightly formatted information.

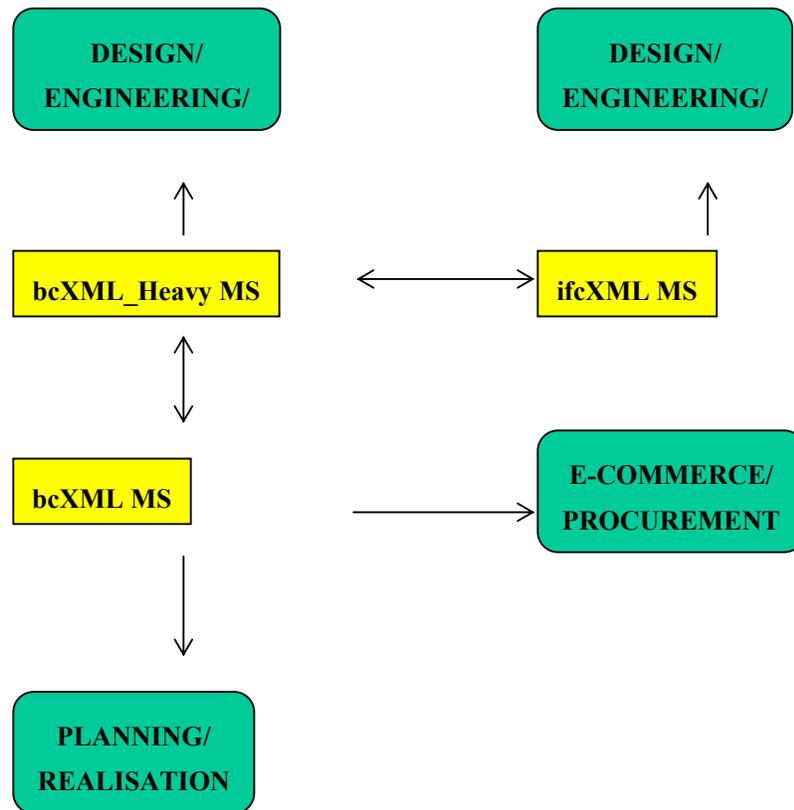


FIG. 2: The green (or dark grey) boxes with rounded edges represent information in certain application worlds, like eCommerce etc. The yellow (or light grey) boxes represent the meta-schemas of the different XML vocabularies and their relations

Typical eCommerce applications requiring transactions are best served by bcXML. Typical design/engineering applications are best served by ifcXML. Applications that combine aspects of eCommerce and design, like Computer Aided Selling of facilities are best served by bcXML.

In parallel with eConstruct UN/CEFACT executed a very interesting project, called ebXML (electronic business XML) that, though focusing on much more advanced industrial sectors, targets many of the transaction-, security-, contract- and business-related requirements of BC. As the ebXML project and its successors are still moving forward, eConstruct only could manage to provide a small bridge between bcXML and ebXML, i.e. re-use of the ebXML “Context” mechanism. This Context mechanism, implemented in the meta-schema, makes it possible for example to dynamically change between native languages, i.e. an order message defined in English is received in French and a French answer is again received in English, or between different views, i.e. the information required by a structural engineer differs from that of a painter.

## 6. DICTIONARY AND TAXONOMY

One of the most important components of the solution concept is the mechanism for capturing construction semantics, i.e. the neutral Building and Construction terms and object definitions. For that purpose eConstruct uses a building related taxonomy developed by the STABU in the Netherlands called the LexiCon (LexiCon, 2001). The LexiCon consists of a structured set of terms and definitions, containing lexical, construction related information about building products, materials and services.

The LexiCon can be seen as a 'neutral' set of building object definitions, here called a *taxonomy*, which can be used to transform one object definition into another object definition. Though eConstruct uses the LexiCon, other (non-neutral) taxonomies are being developed in related BC sub-sectors. This phenomenon, which might be called chaos, is unavoidable in the fragmented BC industry. The problem that eConstruct therefore faces is how to develop a Communication Technology that can be used despite all the many different object definitions (taxonomies)?

Figure 3 shows the position of the LexiCon meta-schema in the two-level architecture presented in figure 2. The LexiCon is an example of a BC taxonomy that provides BC semantics (object and property names and definitions). Taxonomies such as the LexiCon are in itself not part of bcXML. They only have to follow certain rules before they can be used to upgrade plain XML with BC meaning.

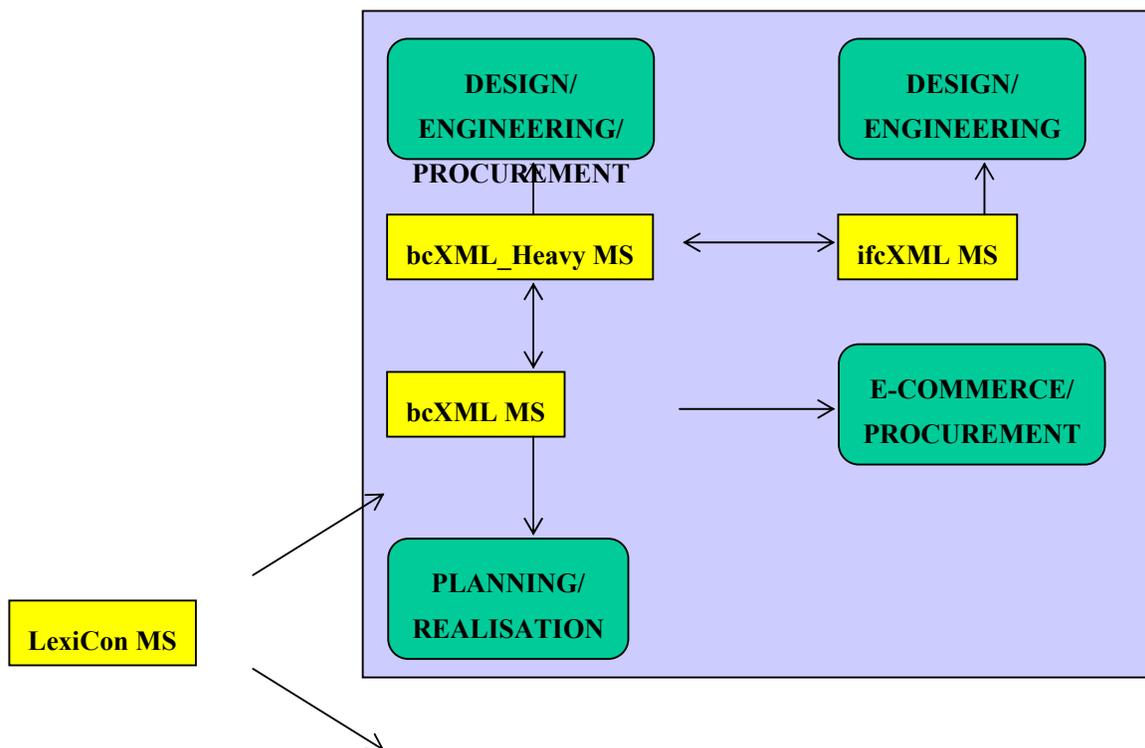


FIG. 3: The LexiCon provides the neutral BC semantics

The advantage of this architecture is of course that every taxonomy builder is free to define the structure of his choice and still is able to provide its users with the required interoperability.

## 7. BCXML-HEAVY

As an example, figure 4 below shows one of the diagrams that defines the bcXML\_Heavy meta-schema in UML.

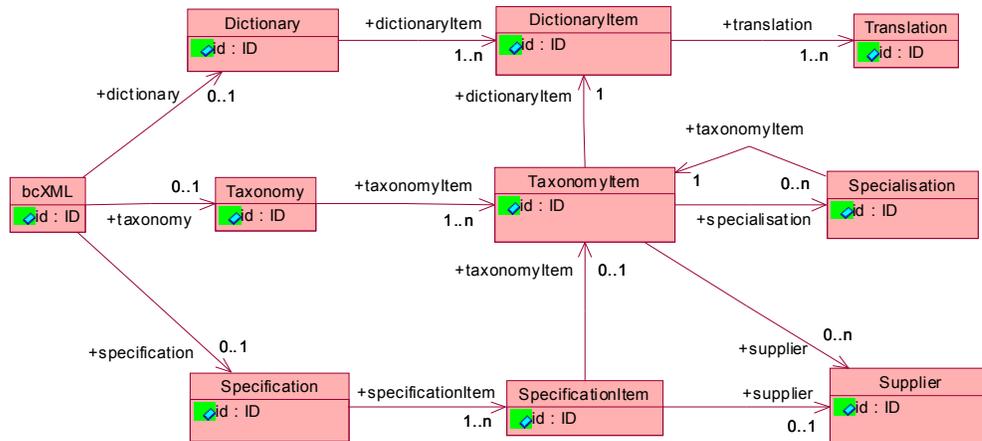


FIG. 4: Content of the bcXML package, i.e. the main classes and their associations

**Dictionaries** are collections of object and property terms and descriptions. As an example eConstruct develops a "bcDictionary" which can be utilised for pan-European bcXML communication scenarios because it is multi-lingual.

**Taxonomies** are the chosen solution for 'managing meaning' in eConstruct. Meaning (semantics) in BC is quite a problem. Think about the thousands of items (objects, properties) that play a role. All items within a taxonomy are related with terms from a dictionary. Taxonomy items are related via Specialisation and interconnection (i.e. objects have properties). eConstruct will also make a start with a public Taxonomy (referencing in this case the bcDictionary).

The model also shows the possibility to define **Specifications**, sets of objects and properties with units and values. Specification is done according to a specific taxonomy and/or dictionary. The specification can be done from the demand-side ("request specification") or the supply-side ("catalogue").

When the bcXML meta-schema is implemented and applied in a communication, the resulting bcXML, called bcXML\_Heavy, looks like the text presented in Figure 5.

```

<bcXML:SpecificationItem id="_1003">
  <item>
    <bcXML:PropertySingleValue id="_1001_original">
      <name>Eigenschap Aantal Wielen</name>
      <nominalValue>

```

```

        <Text>4</Text>
    </nominalValue>
</bcXML:PropertySingleValue>
</item>
<taxonomyItemSelect>
    <bcXML:ExternalTaxonomyItem id="_1002">
        <language>nl</language>
        <externalTaxonomyName>Undefined</externalTaxonomyName>
        <externalTaxonomyItemName>
            undefined
        </externalTaxonomyItemName>
    </bcXML:ExternalTaxonomyItem>
</taxonomyItemSelect>
</bcXML:SpecificationItem>
<bcXML:SpecificationItem id="_1006">
    <item>
        <bcXML:PropertySingleValue id="_1004_original">
            <name>Eigenschap Achterdeur Aanwezig</name>
            <nominalValue>
                <Text>YES</Text>
            </nominalValue>
        </bcXML:PropertySingleValue>
    </item>

```

FIG. 5: Random piece of a bcXML\_Heavy specification

As can be seen from the example bcXML\_Heavy is both very powerful and not really very compact and readable. Its complexity stems from the desired design/engineering compatibility, which often requires quite complex property definitions.

## 8. BCXML

Next we will concentrate on the second development stream that produces bcXML (a “lite” version of bcXML\_Heavy) and that implements a set of simple Use Case scenarios.

The second stream picked up the recently published XML Schema definition (XSD, 2001) and developed a more XML spirited bcXML. Though the bcXML language is (still) somewhat less powerful than bcXML – the limitations directly stem from the limitations of XML Schema – bcXML is indeed very readable and easy to use by programmers that want to apply XSLT to derive filtered and transformed input to their tools. Figure 6 shows a small piece of bcXML that describes an inner door.

```

<InnerDoor name="MyDoor"
  externalReferences="http://www.bcXML.org/MyDoor.ppt">
  <EAN>
    <SingleValue>1234-abc-4321</SingleValue>
  </EAN>
  <height unit="m">
    <SingleValue>1.9</SingleValue>
  </height>
  <thickness unit="m" prefix="c">
    <Range>

```

```

        <min>6</min>
        <max>8</max>
    </Range>
</thickness>
<rotation>
    <SingleValue>left</SingleValue>
</rotation>
<Hinge name="h1">
    <weight unit="kg">
        <SingleValue>0.1</SingleValue>
    </weight>
</Hinge>
<Hinge name="h2">
    <weight unit="kg">
        <SingleValue>0.1</SingleValue>
    </weight>
</Hinge>
<Panel name="p1">
    <External>
        <Fuellung name="f1"
            xtdReuse="ForProperties"
            xmlns="http://www.bcXML.org/2001/Heinze">
            <typ>
                <SingleValue>Klassisch</SingleValue>
            </typ>
        </Fuellung>
    </External>
    <width unit="m">
        <Enumeration>
            <value>0.4</value>
            <value>0.5</value>
        </Enumeration>
    </width>
</Panel>
</InnerDoor>

```

FIG. 6: Example of bcXML. Note the use of meaningful 'tags' like Hinge etc. and the way meaningful properties can be described. Note also the External statement used to import external definitions from a non-standard taxonomy.

How the above bcXML has been derived is explained in the next sections.

## 9. XML TAXONOMY DEFINITION (XTD)

As shown in figure 3, the bcXML meta-schema is totally independent of BC semantics. The meta-schema is therefore referred to as the XTD: XML Taxonomy Definition, see figure 7 for a UML description of the XTD. The XTD is a BC-independent meta-schema used in eConstruct for modelling content-aspects for eCommerce and eBusiness transactions. More information on the eConstruct project and bcXML can be found on [www.eConstruct.org](http://www.eConstruct.org).

This XTD bridges the level of XML Schema Definition (XSD) and the actual industry-sector specific schemas, like bcXML for the description of goods & services between demand and supply in both supply-chains and goods & services development life cycles. It allows for the explicit and semantic definition of objects, properties and their interrelationships.

This XTD meta-schema is seen to be complementary to and/or a potential valuable extension for the more transaction-oriented ebXML modelling work. If combined it allows the coherent and consistent development of schemas defining specification messages (demand and supply) within an ebXML envelope.

The XTD format is closely harmonised with the ISO/DPAS 12006-3 effort, which provides us with good interoperability with specification efforts like undertaken in the LexiCon (LexiCon, 2001). The taxonomy data used within eConstruct is mainly exported from the current LexiCon database maintained by STABU.

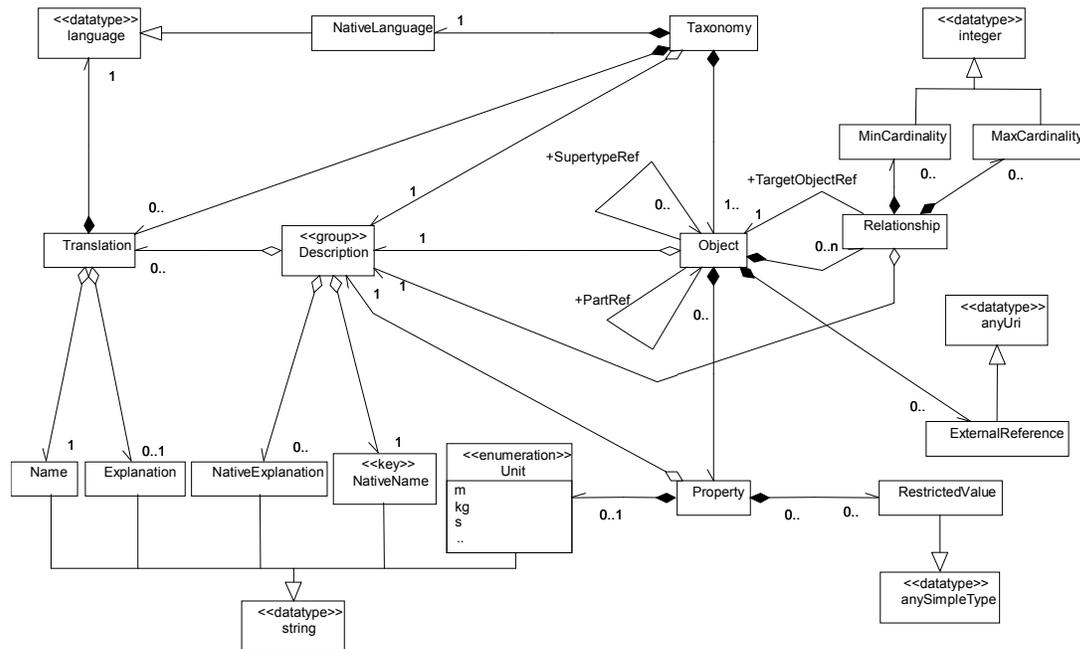


FIG. 7: XML Taxonomy Definition (XTD) meta-schema.

Taxonomy Concepts that are supported are:

- Multi-lingual data.
- Objects.
- Object specialisation.
- Object decomposition (including maximum number of occurrences for different types).
- (Assigned) Properties.
- Units.
- Restricted Values (i.e. "allowed values").
- General Relationships
- External references.

Taxonomy Concepts that are not supported are:

- No separate object-independent properties. Assigned Properties only.
- No complex (aggregate) properties (atomic properties only).
- No explicit placement of parts in wholes (other than via Properties).

The XTD is used to define the objects and their properties. This is the sole purpose of this data format. The actual data (requirement specifications, catalogues) is placed in an XML format modelled after the data in the

XTD file. That format is purpose-built for easy communication according the taxonomy in the XTD file. This two-level modelling process integrates well in the multi-level ebXML developments for the transactions.

This is achieved by this four-level method:

- The start is the XTD format definition (.xsd).
- The taxonomy is expressed in that XTD format (.xml).
- From that XTD-formatted taxonomy a purpose-built format definition is generated (.xsd).
- A requirements message or a catalogue containing the actual data is made, expressed in the format generated in step 3 (.xml).

The result is a lean, small, readable format (Figure 6) without unneeded complexity with the right amount of expression power to be able to create catalogues and requirement messages according to the definitions in a certain taxonomy. The possibility to mix definitions from multiple taxonomies provides the needed flexibility. It is not difficult to see that only simple XSLT-processing is needed to visualise this nicely to an ordinary end-user. The key-factor for obtaining the required simplicity is the strict division between Definitions (all taxonomy related items) and Specifications by generating a purpose-built schema from the taxonomy data (which is fully automated). Specifiers and catalogue builders often do not even ever see the XTD format.

As can be seen from figure 6, the virtue of bcXML is that it closely follows the XSD (XML Schema Definition) and produces simple self-explanatory XML. Its ease of use is demonstrated by its rapid application in the Visualiser, as shown in chapter 12, which heavily relies on XSLT transformations. Picking up and filtering the required objects and their properties is what in the near future Internet-programming is all about.

## 10. THE REFERENCE ARCHITECTURE AND THE PROTOTYPE

The eConstruct framework used to test and evaluate the benefits achieved using bcXML is shown in Figure 8 below. It is composed by a software prototype that is intended to simplify the use of the bcXML functionality and a set of “client applications” (bcXML compliant) that, indeed, take advantage of the prototype and fully exploit these functionalities.

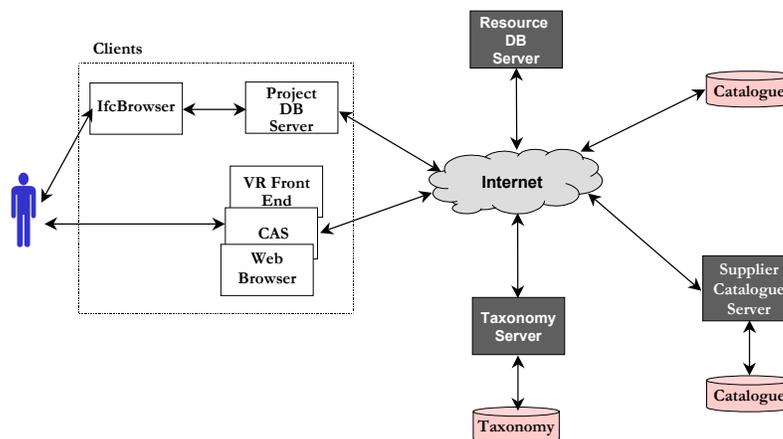


FIG. 8: The eConstruct framework – a general overview

The kernel of the eConstruct Prototype is composed of three main modules, namely: the *Resource DB Server* (RS), the *Taxonomy Server* (TS), and the *Supplier Catalogue Server* (SCS). Due to functional reasons, the RS is decomposed in two modules: the *Resource DB Server Front-End* (RSFE) and the *Resource DB Server Back-End* (RSBE). The RSFE deals with all the human interface related issues. The RSBE supports the searching process

based on the bcXML compliant queries received from the RSFE and from the applications as well. The TS takes care of all the issues related to bcXML-compliant taxonomies (e.g. the bcBuildingDefinitions taxonomy). The SCS is the “gatekeeper” that provides access to the bcXML compliant Suppliers’ catalogues. It is important to mention that the SCS is being offered in “two flavours”: (i) the *Lightweight Server*, which is intended to deal with the bcXML compliant catalogues containing simple products; and (ii) the *XML Server* which works with the Project DB Server in order to support the heavyweight version of Prototype. This paper only considers the Lightweight Server, here referred as SCS.

The Client applications pointed out in the framework are the following: the Computer Aided Selling system (CAS), the *Virtual Reality Front End*, the *Project DB Server*, and the *IfcBrowser*. Besides that, the ordinary *Web Browser* is intended to support all the human-oriented interactions targeting the Taxonomy and the products existing in the suppliers’ catalogues. The Figure 9 shows a more detailed view of the Prototype.

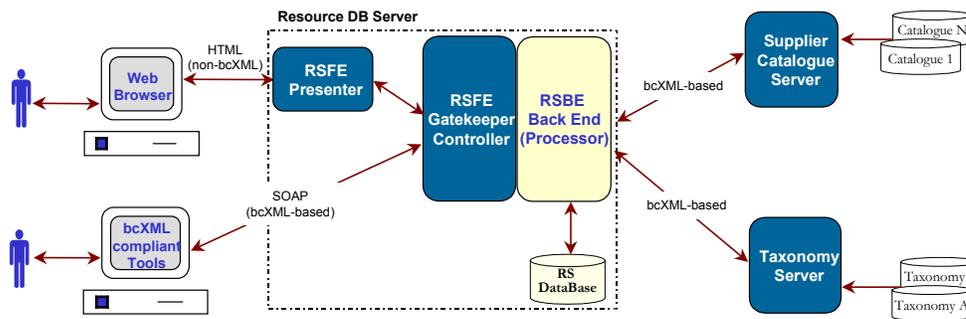


FIG. 9: Detailed view of the framework

## 11. BCXML USE CASE SCENARIOS

This chapter will describe the current set of bcXML supported Use Cases, which are:

- bcXML supported Publishing
- bcXML supported Buying
- bcXML supported Procurement
- bcXML supported CAS (Computer Aided Selling)

In the next sections these Use Cases will be elaborated further.

### 11.1 bcXML supported publishing

Publishing supply-side information often is a difficult task. The increasing complexity of catalogues makes them subject to change at an ever-increasing rate. Reaching a wide enough audience is another problem, especially when the European dimension (with its different languages, amongst other difficulties) is taken into account. The next sections will discuss how to build and publish catalogues with bcXML.

#### 11.1.1 Building a Catalogue

When a vendor decides to publish his product information using bcXML and bcXML-compatible publishing tools, he first has to define his information according to bcXML’s way of publishing.

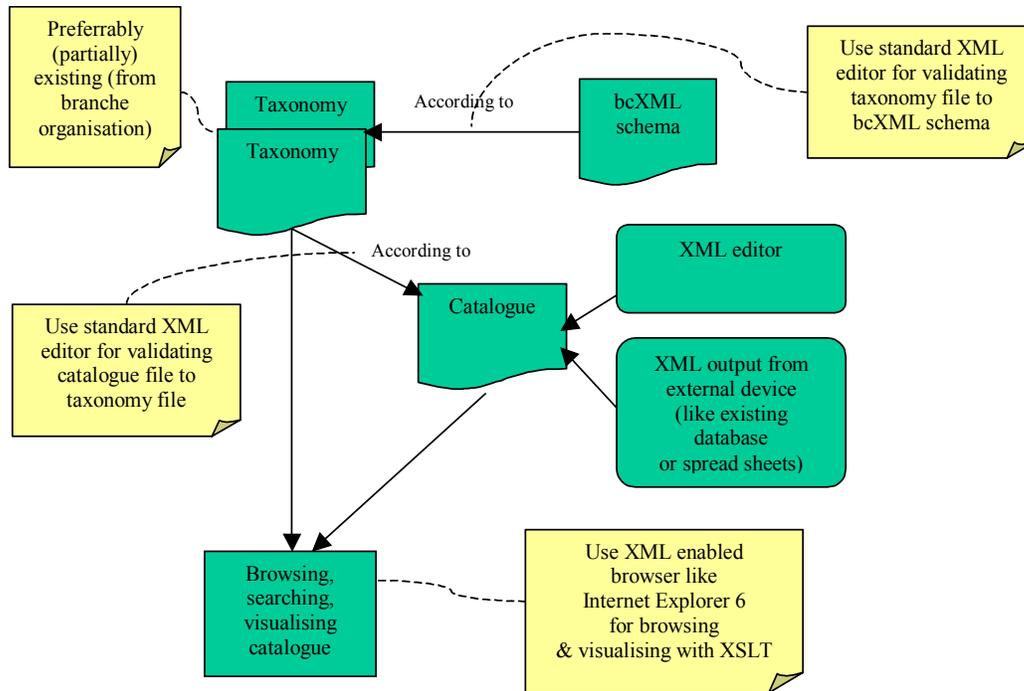


FIG. 10: Catalogue Building

The best start is to search for available taxonomies (definitions of objects and properties) that can be used for publishing. Using already existing taxonomies as the LexiCon has many advantages as it will save a lot of time and effort in a later stage.

After choosing the taxonomy it can optionally be extended by writing one's own small taxonomy for the things not covered by the existing one. This can be done by using an XML editor that automatically checks the structure of the resulting Taxonomy file against the XTD. Or, if available, a specific taxonomy-building-application can be used. Generally, a publisher that doesn't publish unfamiliar building components has a better chance of finding a usable taxonomy, developed by for instance a classification institute or a branch organisation. If the company produces very specific, unique components however, it will probably result in the need for extending an existing, or in totally developing an own, taxonomy.

With the Taxonomy file(s) available, the next step is to create the Catalogue file. This can be done by an XML editor that automatically ensures the file being created to be according to the Taxonomy file. Alternatively, a more sophisticated catalogue builder application can be used, like the Internet-based one being created by eConstruct.

Generally, for information publishers or vendors with a limited amount of different products, a standard XML editor or a standard catalogue builder application as developed within the eConstruct consortium will do the job.

However, for a very large amount of different products (> 100) that are already stored electronically, it will pay off to export the data automatically to the Taxonomy compliant format.

At this stage, bcXML-compliant Taxonomy and Catalogue data are available. This means that the owner of the data has the benefit of multi-lingual capabilities (when the multi-language information has been made available in the Taxonomy file), creation of graphical representations of the information (using SVG, X3D, etcetera) and all the searching and browsing possibilities of the bcXML infrastructure.

When taxonomy files are already available, for instance developed by branch organisations or the basic ones developed by eConstruct consortium, and these definitions cover most of the catalogue content, these taxonomies empower your catalogue with multi-language and graphics (2D/3D) capabilities, only the link between the catalogue and the taxonomy is needed. Then there is no need to define the multi lingual information or the visualisation capabilities oneself.

To summarise:

- For information suppliers that have: a) a small amount of different products (less than 100) and b) no exceptional objects (so taxonomies are available), they can publish their catalogue data using bcXML by entering their product data with a standard XML editor or catalogue builder application in combination with the existing taxonomy. This results in a multilingual, multimedia powered, searchable and findable catalogue.
- For information suppliers that have a larger amount of different products (more than 100) it pays off to put their information in bcXML compliant way by exporting their information in the bcXML format from their existing applications/databases.
- For information suppliers that have products that are mostly not covered by existing taxonomies it is more complicated. They should build their own taxonomy (or make sure the data they need is included in existing taxonomies). This can be done with an XML editor or taxonomy-builder application. However, to achieve multilingual and visualisation capabilities the taxonomy should be provided with the information needed for this.

### **11.1.2 Publishing your catalogue information**

There are four different ways to publish the now-ready catalogue information:

- On paper
- Offline electronically (Palmtop, CD-ROM, send by e-mail)
- Online
- Within the eConstruct framework.

For cases 1 to 3 catalogue information has to be produced in a non-bcXML format, like PDF or html. The fourth case is the only one in which bcXML information can be sent natively, which provides extra functionality.

### **11.1.3 Publishing catalogue on paper and off-line electronic publishing**

For publishing the catalogue on paper it is needed first to transform the bcXML catalogue and taxonomy files to an output format suitable for printing. Likewise, to publish it electronically the information also has to be transformed into another format. This can be achieved on a simple desktop PC by using a freely available XSLT parser and PDF generator. Those tools use style sheets that contain both formatting information as well as the desired output structure of the catalogue to generate a PDF file containing the catalogue. This PDF file can be visualised by a free-available PDF reader. This catalogue is ready to print or send by e-mail as attachment. Also it can be used for further (word-) processing.

It is possible to use a browser like Internet Explorer 6 with built-in XSLT processor to access the bcXML data with all functionality provided by the XSLT style sheet. Seen this way it is possible to publish our information electronically by simply distributing our bcXML files. However, since most users' programs currently are not XSLT enabled, we need to make our information accessible for those programs also.

It is also desirable to generate formats other than PDF. One form would be the HTML for distributing the catalogue on CD-ROM or by e-mail attachment. And another would be a format suitable for downloading to Personal Digital Assistants, like Palmtops. This can be done locally in the same way as the above generation of the PDF file, only now with formatting information applicable to the just mentioned output media. Apart from the PDF generator, also a SVG generator is needed to generate the graphical information in SVG format (Scalable Vector Graphics, an XML 2D picture format).

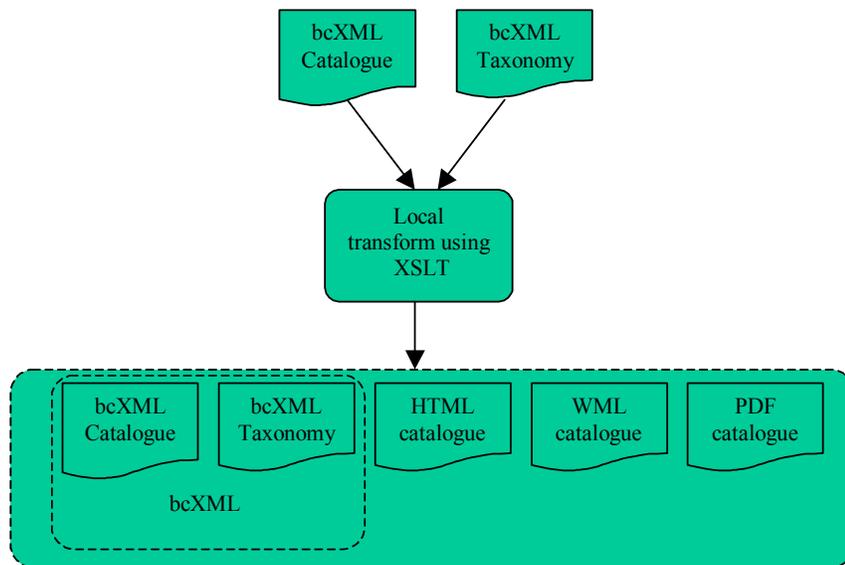


FIG. 11: Local generation of publishable content formats using XSLT (WML = Wireless Mark-up Language)

#### 11.1.4 Online Publishing

Publishing the same information online means the involvement of a web server. Just like in the offline-scenario, the data has to be accessible to all users, both with XML compliant (XSLT enabled browser) and non-XML compliant programs. Therefore both the bcXML-formatted catalogue and taxonomy as well as the exported HTML, PDF and PDA suitable mark-up versions have to be accessible.

Putting the bcXML- and other files online on a standard HTTP server is enough. XML-compliant browsers will be able to access the information with full functionality. Non-XML compliant programs, since they have only access to the catalogue by means of the pre-generated static catalogue files, lack some functionality because of the conversion.

Putting the bcXML data on a web server makes that data available to XML-capable browsers. The information contained within those files must be made available to non-XML programs, which means that this should be done on the server side. By using an XSLT-capable server combined with the appropriate style sheets to visualise and query the bcXML-contained information it is possible to provide non-XML programs with almost full bcXML functionality. Note that it is even not needed to have the XSLT processor and the bcXML files on the same server. An XSLT enabled server with appropriate style sheets can be used to visualise bcXML files residing on another server.

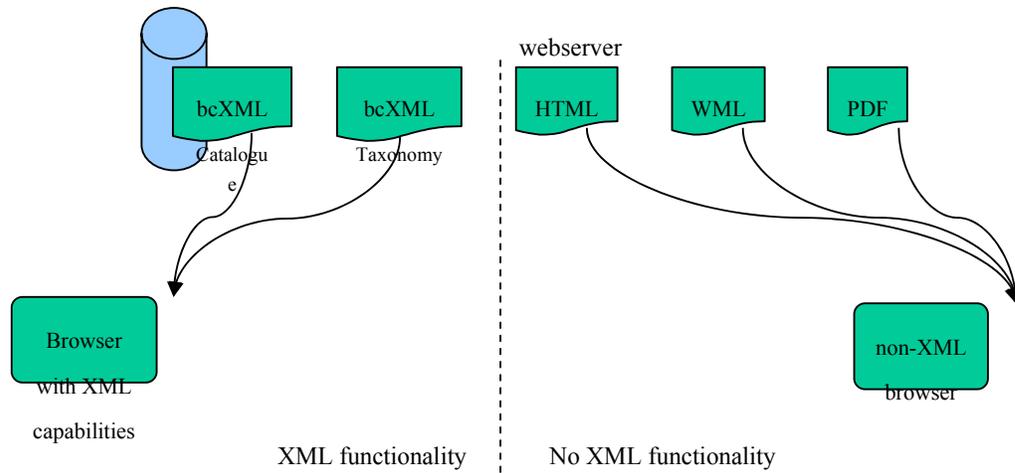


FIG. 12: First scenario for publishing using a standard HTTP server. Both suited for XML and classical clients

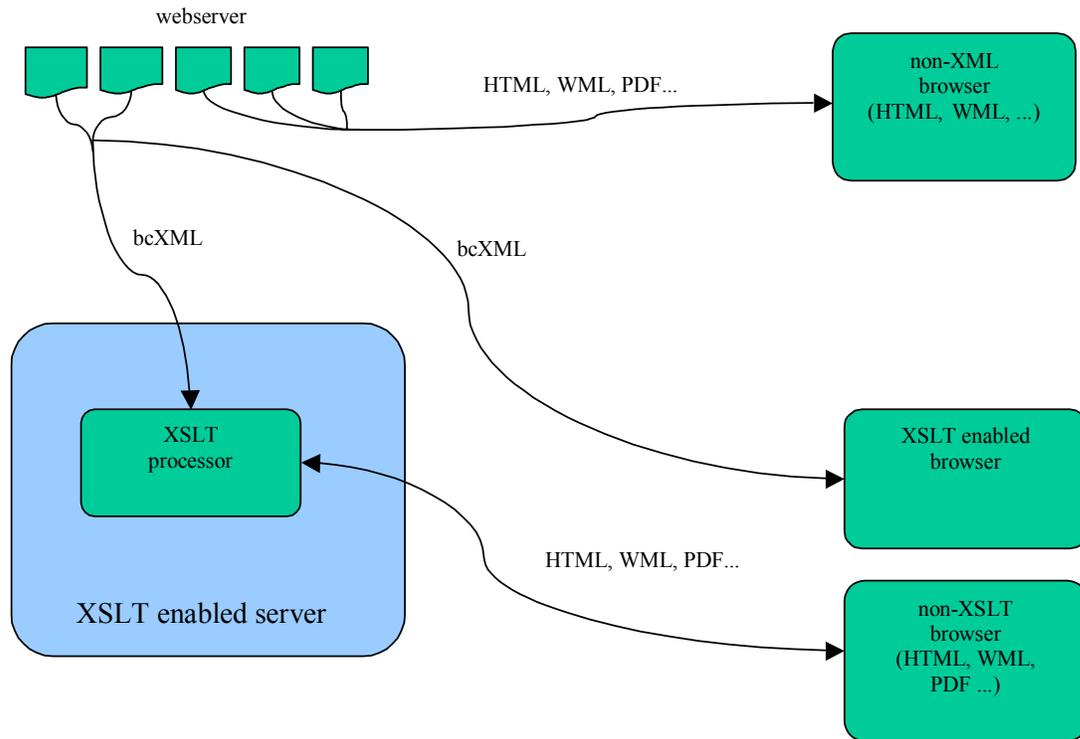


FIG. 13: A web server which publishes both bcXML and html/pdf files and another web server which transforms bcXML to html/pdf

### 11.1.5 Publishing bcXML – empowered by eConstruct publishing framework

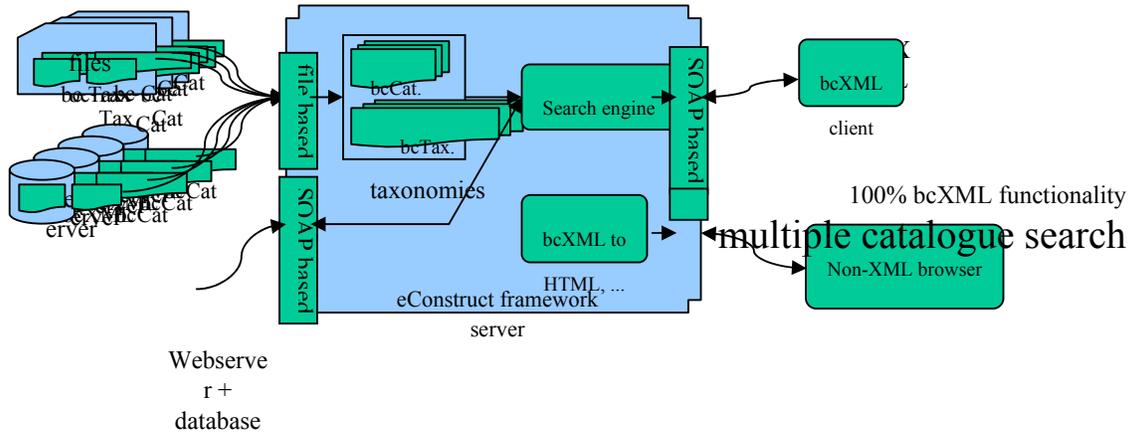


FIG. 14: Many possibilities to publish the catalogue data within the bcXML framework

The previous sections have shown how to publish the bcXML information on the Internet, both in bcXML format and in html, PDF, etcetera formats. But this is limited to one-on-one contact, with a customer specifically contacting the server where the bcXML data was published. Publishing the data using the eConstruct framework introduced in chapter 11 adds the following:

1. Integrated information exchange with the chosen taxonomy, the taxonomy is automatically available for translations, definitions, graphical information, etcetera about the objects and their properties used in the catalogue.
2. The added possibility for automated information gathering using the framework's well-defined XML-based services.
3. Use of the search functionality (in multiple catalogues) provided by the framework.
4. Possible storage of the catalogue (if the company does not store the data on a dedicated catalogue server or stores it simply as a bcXML file on a web server)

Prerequisites for publishing the data within the framework are:

1. The catalogue must be in the bcXML format.
2. The catalogue must be stored in a reachable location (file on a web server, within a company's catalogue server, on another catalogue server).
3. The taxonomy used must be available within the framework (which means that it should be placed within a reachable taxonomy server).

## 11.2 bcXML supported Buying

BcXML supported Buying means the searching, browsing and visualising of the information in a bcXML file with the goal of gathering information needed for the decision to purchase something. This places an emphasis on *quick interpretation* of information to find and select the correct item. This results in the following requirements:

1. Native language support. This is needed to make the textual information easier interpretable. Native language support is not just needed for the user interface, but also as much as possible for the content.
2. *Meaningful* graphical support. Since humans interpret graphical information far more quickly than textual information, graphical representation of the actual content data is a must. And the aim is not

to add flashy pictures or animations to the user interface, but to visualise the real *data* in order to provide real support for interpretation of the data.

3. Context. Since not everybody looks at the same information with the same purpose, it is necessary to exclude information that is not relevant for the viewer. An architect is interested in different things when selecting a product than a constructor.
4. Interaction. The information is presented in a certain way, but by intuitive clicking on graphics or hyperlinks additional information should be instantly available, information probably initially hidden by the chosen viewpoint/context or information available from external sources.
5. Selection. The possibility to *browse* through a catalogue is not enough, the ability to search for catalogue items matching certain criteria is needed.
6. Availability (in time and form). The client needs to be provided by real value and service in order for him to use the information. This mandates the continuous availability of up-to-date information. Also the form of the information is important, it should be accessible in many formats. Printed on paper, accessed by wireless WAP client, catalogue information available in a Personal Digital Assistant, off-line on CD-ROM and online over the Internet. The ability to render information from one and the same bcXML information source is important.

The bcXML supported buying application developed in eConstruct can best be illustrated by figure 15 below. The Visualiser is a simple tool that only requires some freely available plug-in extensions to the browser. The input is a bcXML file that describes a component, in this example a reinforced concrete beam. The different output formats are a selection of the formats mentioned in the requirements.

- Simple HTML.
- SVG (Scalable Vector Graphics, 2D) with the view context “outside details” and the language context “English”.
- Also SVG but now with the view context “inside details”.
- X3D/VRML

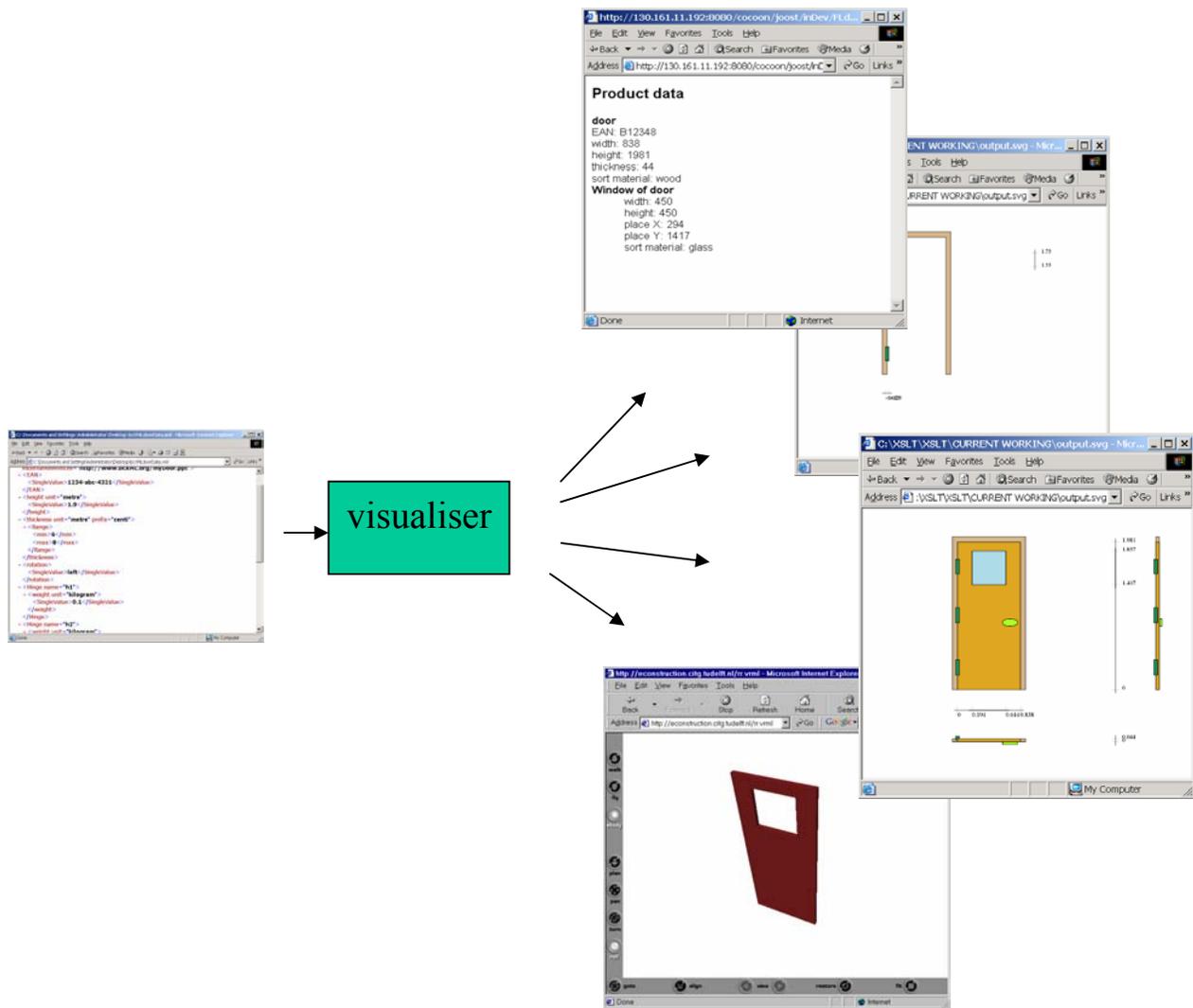


FIG.15: Content defined in bcXML and 4 mark-ups. The unreadable window at the left contains the bcXML example of figure 6.

### 11.3 bcXML supported Procurement

Using bcXML support for project related procurement of components, services and materials (eProcurement) supersedes the current functionality provided by construction-related marketplaces and portals.

The eProcurement application developed in eConstruct will be used to display building models that are held in a so-called Project Database. Information in the building model will be used as the basis for making requirement messages, for products like doors, windows and their associated ironmongery, to the Resource Database Server. Suitable catalogue solutions will then be searched for and presented to the end-user. Once a suitable product has been selected from the catalogue, the product information can be stored in the Project Database building model as occurrences of the specific catalogue information.

A typical end-user scenario is outlined as follows:

Display the building model for example using the IfcBrowser (ifcBrowser, 2001). The user will select one or more Doors in the building model. The appropriate generic catalogue item from the “bcBuildingDefinitions” Taxonomy will automatically be selected. The user will then select a suitable sub-type of Door if appropriate.

The user interface will then display the properties that are associated with the generic element in the Taxonomy. The user will then be able to automatically extract from the building model Door element some values for a number of the properties. Not all the properties will need to have values. Some of the values will be defined as a range and some as enumerations.

The submitted query will then return a list of possible specific catalogue items that meet the user’s requirements. The catalogue items could be from a number of different companies – so more than one catalogue may be accessed.

The user will choose one of the possible Doors and all the associated information about the specific Door will be stored in the “Project Database” as an occurrence of the specific catalogue item.

The following additional information that is related to the possible catalogue item will also be provided to the user:

- Catalogue price.
- Availability, if a stock item
- Additional property information that further describes the specific catalogue item.
- Etc.

#### **11.4 bcXML supported CAS**

One scenario that fully describes the power of the *e-procurement* services provided by the Prototype takes into account the CAS system as the bcXML compliant client. The CAS system considered here supports the sales process of real estates (apartments and houses) to end customers by using a specific application, which allows sales persons to easily present the objects to be sold to their customers in an attractive manner based on always up-to-date product and catalogue data.

The CAS user who is responsible for setting up the catalogue data for a certain project wants to add a special kind of door to the CAS catalogue that fits his ideas in design and quality. Therefore, in the *CAS Catalogue Definer* (a CAS component that supports the management of the CAS internal catalogues), the user opens a dialog where he can select between different possible catalogue element types. For illustrative purposes only, let us consider that the user selects “*Generic Door*” and he/she is provided with a mask where he/she can enter the search criteria that describe the kind of door is wanted to be added to the catalogue. The *CAS Catalogue Definer* creates a bcXML query using the given data and sends it out to the Prototype (using the SOAP protocol). The CAS Catalogue Definer waits for the response and after having received it, it will import the result of the query and display the list of door set specifications matching the search criteria. Finally, the user may select one or more of these data and put it into the CAS catalogue via Drag&Drop. This will create a catalogue item specific to the CAS object format, so some of the original properties might no longer be seen from inside the CAS.

## **12. EVALUATION, CONCLUSIONS AND RECOMMENDATIONS**

Though implementation, testing and demonstration is still on-going, it seems that most of the expectations (section 4) at the start of eConstruct have been fulfilled, except of course the expectation that the Building and Construction industry will pick up the results and apply them to their benefit. There is a Communication Technology (bcXML and related tools) and a partial neutral building taxonomy (LexiCon) available. The particular demonstration domain elaborated in eConstruct is Doors and Precast elements. Other domains are only sparsely occupied.

The results of the eConstruct project are developed in two streams. The first stream uses available technology to develop bcXML\_Heavy and applies it in a number of ‘heavy duty’ Use Cases. BcXML\_Heavy shows to be a

very powerful CT that can be used in many application areas where procurement aspects are being combined with design/engineering aspects. The resulting bcXML\_Heavy language however is not particularly human readable (though of course it can be marked-up for human eyes).

The second stream picked up the recently published XML Schema definition (XSD, 2001) and developed a more XML spirited bcXML, simply called bcXML. Though this bcXML language is (still) somewhat less powerful than bcXML\_Heavy – the limitations directly stem from the limitations of XML Schema – bcXML is indeed very readable, flexible and easy to use by programmers that want to apply XSLT to derive filtered and transformed input to their tools.

The virtue of bcXML is that it closely follows the XSD (XML Schema Definition) and produces simple self-explanatory XML. Its ease of use is demonstrated by its rapid application in the Visualiser, as discussed above, which relies heavily on XSLT transformations. Picking up and filtering the required objects and their properties is what in the near future Internet-programming is all about.

Two other eConstruct results are worth stressing. The first result is that bcXML can be used to communicate in different languages. A request in German is received in France in French, and the French answer is again received in Germany in German. The same can for example be done with the Greece language and alphabet. This feature certainly will help the European BC industry to become more united and competitive. Other languages, also with other alphabets, can be added in the future.

The second result is the fact that bcXML can be used to communicate *over* different taxonomies. This feature makes it possible to forget about the information chaos and still obtain results. No difficult standardisation processes are required.

Besides all the XML-related tools, bcXML's two main pillars are the XTD and the LexiCon. Both XTD and the LexiCon will need further development after eConstruct has been finished. A project proposal for XTD – ebXML harmonisation has been proposed to CEN-ISSS. Further development of the LexiCon content is considered by the International Classification Institute Society ICIS. Groups interested in taxonomy building are well advised to study the eConstruct approach. Also other taxonomy builders should consider the added value that bcXML can provide to themselves and their clients.

Another interesting remark is directed to the vendors of larger construction assemblies like roofs, frames, or facades, or complete structures like office buildings or housings. Computer Aided Selling systems that support the automatic linking of external catalogues and such have great potential because, without prior efforts, they are able to find and use the best components, materials and/or services available on the market, also – if required – on the international market.

Yet another recommendation is to include more language support, also for the new Eastern European countries. Those countries are well placed to find an interesting role in the supply-chain of construction products, materials and services. A new European initiative in this area is highly recommended, not only in the building area, but also in related areas of construction and civil engineering.

Further research is recommended into eBusiness and specifically in the use of bcXML in collaborative construction. In the future construction consortium building will become an important factor for success. Construction companies should increase their skills to quickly participate in a new project team. This research area overlaps with the UN/CEFACT ebXML goals.

Finally an interesting R&D area is the application of VR *project* information front-ends. In the future all parties involved in a construction project – including the public and the authorities – have access to a Virtual Reality project world from every place PC, notebook or PDA connected to the Internet. Like the illustration above in figure 15 users with different background and authority can see different things, if necessary in different native languages. Browsing and interacting with the project world is possible, both by mouse-control and/or by speech-control. This functionality supports a whole new crop of Internet applications that should be interesting for the

R&D community, the application vendor community, the Building and Construction industry, the general public and the authorities.

### 13. REFERENCES

Econstruct (2001). eConstruct homepage <http://www.eConstruct.org>

XSD (2001). W3C XML Schema <http://www.w3.org/XML/Schema>

Tolman F, Stephens J, Steinmann R, Van Rees R, Böhms M and Zarli A (2001). bcXML, an XML vocabulary for building and construction, *ECCE conference Finland 2001*

BcXML(2001). eConstruct homepage <http://www.eConstruct.org>

LexiCon (2001). Stabu homepage <http://www.stabu.nl>

Woestenenk K (2001). The LexiCon: a matter of concepts, *IBIC conference Portugal 2001*

IFCbrowser (2001). TNO Bouw homepage <http://www.bouw.tno.nl>

Zapthink (2001). The pros and cons of XML <http://www.zapthink.com/reports/proscons.html>

SVG (2001). Scalable vector graphics <http://www.w3.org/Graphics/SVG>

X3D (2001). X3D FAQ <http://www.web3d.org/TaskGroups/x3d/faq/>

XSLT (2001). Resource site for xslt, xsl & xml <http://www.xslt.com/>

IfcXML. IFCxml [http://cig.bre.co.uk/iai\\_uk/IFCXML.htm](http://cig.bre.co.uk/iai_uk/IFCXML.htm)

IAI (2001) Links to IAI and IFC <http://cic.vtt.fi/links/ifc.html>

### 14. APPENDIX A: ECONSTRUCT PROJECT

The eConstruct consortium includes the following partners and their roles:

Taylor Woodrow, UK, Project Co-ordinator  
<http://www.twc.co.uk>

Betanet SA, Greece, End-User  
<http://www.betanet.gr>

Netherlands Organisation for Applied Scientific Research (TNO), R&D, Technical Project Management  
The Netherlands [http://www.bouw.tno.nl/about\\_us/organization/mit/bpi](http://www.bouw.tno.nl/about_us/organization/mit/bpi)

Centre Scientifique et Technique du Bâtiment (CSTB), France, R&D  
<http://www.cstb.fr/frame.asp?URL=english/default.htm>

Nemetschek AG, Germany, ICT vendor  
<http://www.nemetschek.de>

EPM Technology AS, Norway, ICT vendor  
<http://www.epmtech.jotne.com>

Stichting Standaardbestek Burger-en Utiliteitsbouw (STABU), The Netherlands, Specification  
<http://www.stabu.nl>

Delft University of Technology, The Netherlands, R&D, Technical Project Management  
<http://cti.ct.tudelft.nl>

The requested project duration was three years, however, the final contract was agreed for a two-year project. Additional information about the European IST 10303 eConstruct project can be found at [www.econstruct.org](http://www.econstruct.org).

Several public domain deliverables can be freely downloaded. Also several software components will become available in the public domain.

*this page is blank*