Classification: Restricted

IST-1999-10303

D601

Virtual Reality Information Front-end

(multi-media enhanced bcXML communication)

Authors:

Frits TOLMAN	TU-Delft
Joost FLEUREN	TU-Delft
Reza BEHESHTI	TU-Delft
Reinout van REES	TU-Delft
Jeff STEPHENS	Taylor Woodrow

Distribution	*Draft1*	*Draft 2*	*Issued4*
Date	23-03-01	23-04-01	11-09-01
BETANET		✓	✓
CSTB		✓	✓
EPM		✓	✓
NEM		✓	✓
STABU		✓	✓
TNO		✓	✓
TUD	✓	✓	✓
TW		✓	✓
WWW	✓	✓	✓
CEC			✓

Deliverable Reference		
WP WP6	*Task* T6100	*No.* D601
Status Draft/Issued/Revised Issued		*Rev.* 4
Date 11-09-2001		Release to CEC Sep 01

Document Control Sheet

Revision	*Status*	*Page Nos.*	*Amendment*	*Date*	*By*
1	Draft		For comment	23-03-2001	TU-Delft
2	Draft		Full document	24-04-2001	TU-Delft
3	Draft		Conclusions updated	20-08-2001	TU-Delft
4	Issued		Final update Issued to the Commission	11-09-2001	TU-Delft

Table of Contents

Table of Figures

1 Summary

WP6 focuses on multi-media extensions of bcXML - often a picture says more than a thousand words. Emphasis is on graphical formats for technical drawing and Virtual Reality (VR) and their subsequent application in Information Front-ends, i.e. the use of VR images or 'worlds' to communicate information. The two tasks defined in the TA are:

- Virtual Reality Information Front-ends (Task 6100)
- Prototype Project Information Front-end SW (Task 6200)

This document describes the result of task T6100: Virtual Reality Information Front-ends.

Section 2 shortly introduces this report

Section 3 details the objectives of the current task

Section 4 gives a set or terms definitions that are used in the rest of the document

Section 5 introduces to the available technology

Section 6 gives an overview of the industry needs

Section 7 explains the scope of the rest of the work (to be) done in WP6 and

Section 8 presents the conclusions

1.1 Introduction

This report presents the results of a research into multi-media enhanced bcXML for application in eCommerce and eBusiness in the Building and Construction (BC) industry. The research has been triggered by the development of the semantically rich bcXML communication language, the upcoming new technologies in multi-media, electronic communication and Virtual Reality, and by the needs of the BC industry.

1.1.1 The technology is there

Due, in large part, to the significant advances in PC hardware and Internet communications that have been made over the last years, PC based distributed VR is approaching reality. While the cost of basic desktop VR systems has gone down by a few thousand dollars, the functionality has improved dramatically, both in terms of graphics processing power and in accessibility.

Not only the continuous increase of the processor speed contributes to this fact, also new generation of low-cost special purpose graphics engines greatly helps. The only bottleneck for large-scale Internet based VR is now the bandwidth of the Internet. Digital networking with UMTS and other standards might here provide the right answer.

In the communication software area the latest web-technology and the development of the semantic web (bcXML in our case) create the opportunity to provide low-cost worldwide meaningful electronic communication. New XML based communication languages also contribute to speed and security.

1.1.2 The needs are real

The main goal of eConstruct is to improve meaningful electronic communication in eCommerce and eBusiness. Communications about construction materials and products often can be much improved if supported by graphical images displaying textures, colours, layout, topology, shape and such.

Added value lies in the use of XML-based visualisation techniques that extend the capabilities of HTML and support 2D and 3D graphics and Virtual Reality. These graphical outputs can help the buyers, beyond the glossy folders, in the selection process; does the product really fit, are the required features available, is the product looking good from different sides? Those, and thousands of other questions can best be answered with multi-media enhanced bcXML.

From the BC industrial point of view the proposed technology helps to bridge the gap between suppliers and customers of materials and components both nationally and internationally. The fact that bcXML is a low cost multi-lingual Internet technology, supports the position of the international supplier community as they can make their product information readily available and continuously keep it up to date. As to the customer community the benefits follow from the increased choice and competition and the fact that procurement processes can be better streamlined.

From the European point of view the results of eConstruct and of WP6 will contribute to the Europification of the BC supply chains and improved co-operation in international projects.

2 Objectives

The XML-based Communication Technology under development by eConstruct (bcXML) mainly focuses on *semantics*. It describes the words and language constructs required for electronic meaningful multi-lingual information exchange in the Building and Construction industry (BC). The bcXML communication language can be used in communications between (electronically connected) humans, between humans and computer applications, or between computer applications only. In this report the focus is only on communications that involve humans.

The nice thing about XML is that it supports the distinction between *content* and *mark-up*. Content is what you want to communicate and mark-up is how you see the content on paper or on the screen. For eConstruct this XML feature is interesting because in multi-lingual communication the mechanism can be used to display the content of a message concurrently in different (European) languages.

Though bcXML and its supporting taxonomies[1] will enormously help the Building and Construction industry in it's effort to change its traditional paper based way of communicating into an electronic way of communicating, in many human communications 'a picture says more then a thousand words'.

The objective of WP6 is to investigate how bcXML communications involving humans can be enhanced by multi-media support, especially by 2D vector graphics and 3D Virtual Reality (VR).

To fully understand the aim and the reason for incorporating the work package in a European project, consider an average one-of-a-kind construction project in the public sector, involving participants, suppliers, authorities, and the general public. In some cases thousands of people, all with daily requests for information. Then imagine a project where people from different nationalities are co-operating to reach the required result, and notice yet another source of miscommunication and consequently of cost of failure.

What-if all parties involved can have access to a virtual world describing the project in some detail? Some detail, not great detail, because the VR world is used to support information retrieval. John Doe can enter this world from his PC at home and look around. He can move to his neighbourhood and view the project from, say, his second floor window. He can scroll forward in time to see when he should plan his holidays because of the piling noise. Think about the many authorities that need up-to-date information about all kinds of aspects, like fire safety, durability, energy, compliancy with building regulations. No wonder that miscommunication and cost of failure plays such an important role.

Improving electronic communication at large is the ultimate goal of bcXML and the work on SVG and VR information front-ends. However this goal is still far away. Before we can start to work on VR front-ends for large-scale projects we first have to understand the technology and implement it for eCommerce. In eConstruct we are developing bcXML that provides us with the right level of semantics. Suddenly we can talk about Wall Elements, or Hollow Concrete Floor Slabs and the Internet 'knows' what we are talking about. How interesting it would be if this promising communication technology can be further enhanced by multi media.

In order to study the question how this can be achieved and which results might be expected, we are looking into the technology and into the short-term needs in eCommerce support. We

[1] The bcXML standard and the bcXML compliant collections of BC terms (Dictionary) and BC object definitions (Taxonomies) provide the industry with electronically available 'neutral' BC semantics.

study how to use bcXML to generate 2D drawings and VR worlds that are needed, and how the user can navigate through these worlds and find information that he needs, but only on a small scale.

As a proof of concept WP6 aims to develop and demonstrate the technology for bcXML supported Publishing and bcXML supported Buying of single components.

WP6 also contributes to the development of the bcXML Reference Architecture and hopes to integrate the work on multi-media enhancements into this tool.

3 Terminology

The purpose of this section is to establish common understanding of terminology related to the task.

3.1 Multi-media

Multi-media currently include *graphics*, like textures, vector graphics, VR and *simulation* of moving objects and changes in time. But also *sound* and *smell*, and in the computer games industry *impacts* play a role. In this study we will only focus on graphics support, though in some areas the application of sounds and speech processing technology will be mentioned.

3.2 Virtual Reality

Virtual Reality is widely used in the computer games industry and in many realistic simulation systems used for training. With the advent of VRML (Virtual Reality Modelling Language) VR also is concurring the Internet. Unfortunately the Internet's bandwidth is still too small to support sharing of realistic large-scale VR worlds, but in the coming years this deficiency will be remedied. The first 10Gbyte network has just been demonstrated. For small-scale VR support the current Internet is reasonably satisfactory.

3.3 VR Information Front-end

Virtual Reality information front-ends displays a VR image or 'world' on the user's screen. The user can look at the VR model, walk around it and move through the virtual world. If the user wants information about a certain object or some feature (like a hole) of an object he clicks on a button that is attached to the object or feature and a window pops up that gives information about the object.

Several enhancements can be envisioned. One particular nice option is to show only the information that is interesting from the users point of view. If the user is, say, a painter he is probably only interested in the material that the object is made of and the external dimensions (square meters) that he has to paint. And if the user is a project manager he is probably only interested in the states of the object: who is/are responsible for the realization and when will they do it or when has the job been finished. This type of extension heavily relies on the functionality of the taxonomy (does it support a view mechanism) and on the ebXML "context" mechanism.

Another interesting application area for this type of technology is the support information sharing and participation processes in large-scale construction of public works like railroads. If all the parties involved have access to the VR world that images the project, and – according to some authorization – are able to find the required information, much miscommunication can be avoided. This type of application requires the possibility to navigate through a sub-set of the VR world, which again requires some dedicated technology.

3.4 bcXML

The bcXML communication language supports communication of a variety of Building and Civil Engineering objects and their properties. Typically there are three types of physical objects:

(1) Off-the-shelf/standard
(2) Configured
(3) Made-to-order/custom designed

Each class contains objects that range from simple to complex. For example of-the-shelf one can buy gypsum plates, steel beams, CV Boilers or window systems. Configured objects range from: stairs and kitchens to system-walls and ceilings, and made-to-order objects range from pre-cast concrete slabs to in-situ concrete frames, or beyond to all the one-of-a-kind projects performed by the industry.

For each category different multi-media support can be envisioned. Producing multi media support for of-the-shelve objects will generally be a task of the supplier. For the configured classes the same might be true, but in many cases it is probably better to generate the graphics from the bcXML file. The simple objects from the made-to-order category probably partially can be treated similar to the configured category, but for complex objects other approaches are required as it will not be possible to describe a complete building model as one huge bcXML file.

4 TECHNOLOGY VIEW

4.1 Introduction

This section provides an overview of the existing technology[2] as seen relevant for the task. Basically four types of multi media support are investigated: (1) VR technology, (2) XML technology like SVG, (3) Parametric Languages and (4) VR generation from geometric modelers or product modelers. Also a few words about speech processing have been added.

4.2 3D Graphics Accelerators

The graphics card landscape is evolving quickly, and the cards available today take good advantage of the advancements being made. Two advancements are particularly interesting for VR users: the **Accelerated Graphics Port (AGP)** and the new **faster 3D chips** (nVidia Riva, Permedia 2, Ticket to Ride 2, 3D Rage Pro, etc.).

- **Accelerated Graphics Port (AGP)**: The accelerated graphics port is a high-speed, point-to-point connection between the system chip set and the graphics chip. AGP provides a high-speed pipeline between the graphics accelerator and the PC's system memory: using an AGP connection, a graphics chip is able to access system memory directly through the system chip set at memory-bus speeds, reducing latency and substantially increasing performance versus standard PCI-memory transfers. The graphics card gains access to system RAM to store and execute texture bitmaps, which allows more detailed textures of unlimited size while speeding 3-D rendering. AGP also helps speed the flow of decoded video from the CPU to the graphics controller. Note that your system's chip set must include support for AGP transactions, and the graphics chip must be built to communicate to the system chip set via AGP protocols, so you won't be able to upgrade your PC to AGP unless you buy a new motherboard. When textures are large, AGP can make the difference between a smooth or choppy frame rate in 3-D rendering. The idea of a separate graphics port to assist the 3-D-processing pipeline is not new - Silicon Graphics has long offered systems that can be configured with up to 16 graphics ports for fully immersive virtual-reality rendering - but it is a very welcome solution in the PC space.

- **Faster 3D chips:** In VR, performance is critical. VEs gave mainstream 3-D acceleration its start, and developers have been adding a sense of realistic depth to their creations for years. However, the addition of a *z*-axis in rendering, as opposed to simply drawing on an *x*, *y*-coordinate plane, requires a lot of sophisticated horsepower. In addition, VR applications contain more complex objects and complex *textures*: bitmap renderings of detailed surfaces (bricks, sand, or transparent water) that heighten realism. To deliver all this and maintain a good frame rate--preferably close to the 30 frames per second of motion video--hardware-assisted acceleration is a must. Like a game of leapfrog, advances in speed for cards in the nascent 3-D arena are coming one after the other. The new chip sets *more than triple the 3-D acceleration* we saw from the previous generation of chips. And this increase in performance is slowing down.

These improvements will reflect on applications soon. Look for more advanced uses of 3-D in *presentation packages*, where its visual impact will make for a more powerful and engaging presentation. Just as interesting are data-visualization and data-mining tools that may help to

[2] Mainly cut and paste from the Internet

make business professionals more productive. Three-dimensional spreadsheets could turn endless columns of numbers into 3-D maps of the country, with sales figures creating mountains or valleys as appropriate, once the numerical data is converted to the 3-D chart form. Correlation of small data sets and their relation to the big picture may become more apparent and not get overlooked.

For *training and education*, workaday PCs will be able to render machine parts and other complex 3-D objects that formerly could only be drawn on expensive graphics workstations. Students could virtually explore faraway places, or walk around a lifelike model of a tyrannosaur in a museum. Or potential home buyers could evaluate a real-estate layout on-screen in a convincing 3-D world.

But the *most prominent delivery vehicle for 3-D content could be the Web*. VRML (Virtual Reality Markup Language) 2.0, the programming standard for 3-D on the Web, enables users to log on to 3-D sites and interact within a 3-D world. Log onto a site, click on an object, and it comes to life; turn it, move it, do what you can't do with a flat image. 3-D interaction could be helpful for everything from shopping to learning about physics on the Web.

There are VRML authoring tools available from several different vendors, and both Microsoft Internet Explorer and Netscape Communicator include VRML 2.0 browsers. While Direct3D support in Windows will enable all users to reap these 3-D benefits via software emulation, users with graphics accelerators able to speed the process along will have a better experience.

4.3 3D APIs

APIs (application programming interfaces), such as those within Microsoft's DirectX Foundation and proprietary schemes from graphics-chip makers, let developers get directly to the display hardware to increase speed. For their part, the graphics chips and APIs support a wide range of special effects that developers can call on: fog at varying densities; *transparency*, to make glass objects or water more realistic; *specular highlighting*, which gives the illusion of light reflected off a surface; lights that resemble spotlights or floodlights that originate from a source and can move with an object; and many more.

While gamers have been reaping the benefits of all this technology, 3-D applications have taken a while to filter through to mainstream business PC users. We're still waiting for the killer application that will make 3-D support a must on every business and home PC. But as the installed base of 3-D-capable PCs reaches critical mass, software developers will start to fill that void.

Pioneers in 3-D productivity applications will likely be software vendors clamouring for recognition among their competitors. The DirectX Media Layer component of Microsoft's DirectX 5a API will make this relatively easy to do. One of its services, Direct3D retained mode, will enable programmers not familiar with 3-D coding to incorporate 3-D objects that move within a 3-D world space, or sit atop a regular 2-D interface. An object like this may be useful in a floating palette of tools, a tutorial, or a wizard that extends the application. There is no a standard 3-D engine in the PC market, primarily since different types of APIs are needed at different levels of development and for different applications. At the high end, OpenGL has become the most accepted standard. OpenGL was originally developed by Silicon Graphics for workstations, but has become very popular on the Windows NT platform. OpenGL is a complex and robust API that is intended for precise applications like scientific visualization and CAD/CAM; however, it is overkill, to say the least, for games and other casual 3-D applications.

Several other individual APIs have sprung up for Windows and DOS: Apple Quickdraw 3D, Argonaut Brender, Criterion Renderware, Rendermorphics RealityLab, and others. Sensing discontinuity, Intel joined the game with *3DR*, an API that was supposed to provide a common framework for application developers and board makers, not to mention a wider market for the Pentium. The 3DR API was beginning to catch on when Microsoft jumped into the ring and convinced Intel to go along with its plans for standard APIs for Windows 95. On Windows 3.x, Microsoft had previously provided *3-D DDI*, a low-level device driver interface for chip makers, but had not delivered a high-level API for application development. Microsoft has a strategy for 3-D but has been slow to bring it to market. First, it bought Rendermorphics to acquire the RealityLab product and announced plans to integrate it within the Windows 95 graphics architecture as its standard high-level API. Microsoft also announced Direct3D, an API layer below RealityLab that will provide the glue between existing APIs, such as OpenGL, and hardware. Direct3D will join the other Windows 95 APIs such as DirectPlay and DirectSound as part of a universal strategy to spur game development. Vendors of 3-D products are indeed optimistic about Microsoft's strategy, although perhaps they're also frustrated. Microsoft has recently released its second beta of Direct3D to developers, and the final version may be released very soon. However, hardware and software vendors alike have been waiting anxiously for Direct3D and RealityLab to finally gel. In the meantime, chip vendors have brought even more proprietary APIs to market as short-term solutions. This delay in software development has given the chip and board makers time to catch up, and there will be no shortage for acceleration hardware when the 3-D software titles start coming to market in force.

Below you will find a brief description of the most promising APIs: **OpenGL, QuickDraw 3D and Direct 3D**.

4.3.1 OpenGL

OpenGL is the only graphics engine we looked at that extends beyond the PC and Mac to many flavours of UNIX. The most interesting feature about OpenGL is that it uses a client/server mechanism to handle graphics processing. A graphics client (e.g., a 3-D application) uses the OpenGL interface and the host's OS to transmit such graphics primitives as vertex co-ordinates, colours, and texture co-ordinates to a graphics server process that runs the OpenGL rasterizer.

The rasterizer performs one or more graphics operations based on the current OpenGL graphics state and generates pixels. The pixel values undergo a final processing stage that includes texture-mapping, fogging, antialiasing, depth testing, and blending. This information goes to the client for display in the application window. Normally, the client and server run on the same system. However, the advantage to the separation is that a low-cost desktop system can transmit OpenGL commands across a network to a high-performance system running the server process, which returns images to the desktop machine.

Compared to QD3D and Direct3D, OpenGL is not a high-level API. Its purpose is to draw objects. Such high level tasks as object editing and file i/o are left up to the application. Instead, OpenGL provides a platform-independent interface to a low-level rendering engine. This interface consists of about 250 drawing commands that let the programmer describe objects and perform a set of operations to produce the final image. You can call OpenGL from a variety of languages, including C/C++, FORTRAN, Ada, and Java. OpenGL supports both an immediate mode and a retained mode for graphics operations. In the immediate mode, an application sends graphics commands to OpenGL, which promptly executes them. This

provides a quick response to any changes to an image, a critical feature for interactive graphics applications.

A number of vendors provide 3-D applications and toolkits for use with OpenGL. open Inventor is a 3-D toolkit that supplies a number of built-in graphics tools such as colour and texture editors. It is available on workstations and Windows. It saves 3-D graphics in an Inventor file format that's a superset of the Virtual Reality Modelling Language (VRML), the industry-standard file format that describes 3-D objects on the Web.

OpenGL supports a wide range of graphics environments that cover the spectrum in terms of cost and performance. on the low end, it provides software-only rendering for desktop PCs. But it can also communicate directly with high-end workstations equipped with visualisation hardware that can draw 50 million polygons per second.

4.3.2 QuickDraw 3D

QD3D is a complete 3-D graphics environment. The top level is an API that enables developers to create and manipulate 3-D objects and to read or write 3-D data to files. This API talks to an extensible graphics pipeline that processes drawing operations. The pipeline in turn talks to a thin hardware abstraction layer (HAL) that provides a device-independent API for game designers. QD3D supports an immediate mode and a retained mode. The immediate mode is identical to OpenGL's in that it's up to the application to supply drawing commands to the renderer. In the retained mode, the object-oriented-programming (OOP) structures maintain the scene's geometry for display and manipulation.

While the immediate mode offers fine-grained control for those applications that need it (e.g., animation, where all the objects change constantly), the retained mode lets you store the scene's geometry in an object database. The retained mode makes it easier to read and write objects, and it enables data structures to be cached for fast display or hardware acceleration. Programmers can opt to use one image mode exclusively, or both as the situation demands it. Where QD3D differs radically from Upends is that it is an object-oriented graphics system. A new instance of an object can inherit characteristics from its class, including the geometry, size, orientation, colour, texture-maps, and lighting, which makes for rapid construction of a scene's objects. This also simplifies the maintenance of each object's information for manipulation and display.

High-level API commands let you create, rotate, edit, illuminate, and apply transformations to objects. A widgets mechanism provides visual " handles" that allow you to edit or scale an object interactively. Thanks to QD3D's object-oriented design, you need no knowledge of the 3-D object's internal structure to perform these actions. Currently, you can use only C/C+ + to make QD3D calls.

Also unlike OpenGL, QD3D lets you read and write 3-D images in a common 3-D metafile (3DMF) format. This format saves not only each object's geometry, but also its lighting and texture-maps. Also, 3DMF enables applications to copy and paste 3-D graphics or drag and drop them into other applications. In March, Netscape, Silicon Graphics, and Apple jointly announced plans to develop a binary file format for VRML 2.0 based on the 3DMF format.

4.3.3 Direct3D

Direct3D, the Microsoft's 3-D API, is a new addition to the company's DirectX interactive-media technology family. Like OpenGL and QD3D, Direct3D supports both immediate and

retained modes of operation. Direct3D's retained mode resembles QD3D in that it offers developers an object-oriented high-level interface to applications for manipulating 3-D objects. After you load an object using one API call, you can rotate and scale it using other API calls. The retained mode provides API calls that read and write a file format that stores 3-D data. This data consists of pre-defined 3-D objects, meshes, textures, and animation sets. This file format lets applications exchange 3-D information and play back animation sequences in real time.

Direct3D's immediate mode is a thin API layer that deals with polygons and vertices. The immediate-mode API passes display lists composed of vertex data and graphics commands known as execute buffers in Microsoft terminology to the rendering engine for processing. This system provides a high-performance device independent mechanism that lets programmers access a system's graphics hardware or tap into hardware accelerators. This mode doesn't offer any object or scene management engines; such functions are the application's responsibility.

The immediate mode is best suited for developers who have an application with its own rendering engine, yet wish to access a system's hardware-acceleration features. Currently, you use C/C + + to access Direct3D calls. Support for Visual Basic is planned.
Direct3D offers a variety of graphics objects. In the immediate mode, these objects include execute buffer, matrix (for transformations), light, texture, material (an object's surface properties), viewport (the screen region owned by the object), device (the hardware managing the screen), and interface. The retained mode provides additional objects based on the immediate-mode objects, including polygon face, mesh, frame (manages the position of all the objects in a scene and their orientations shadow, and animate (used to apply transformations to an object, such as scaling and position).

Direct3D is more like OpenGL in that it provides drawing primitives, but not such basic objects as spheres, cylinders, or cones, as QD3D does. It's not clear at this time whether such objects will be available in a separate library, or whether you will have to construct them from scratch. Also, information on the interactive editing of these objects is lacking.

4.4 VRML

VRML, pronounced "VER-mul", is an abbreviation for Virtual Reality Modelling Language. You might see some references to "Virtual Reality *Mark-up* Language", which is what VRML was called at the very beginning - taking its cue from HTML: Hypertext Mark-up Language - but it's been several years since people realized that our vision could be and ought to be a good deal bigger than simply marking up text to add on 3D capabilities.

VRML is, in the words of the VRML Consortium, "an open standard for 3D multimedia and shared virtual worlds on the Internet."

- **An open standard:** VRML was recognized as an international standard (ISO/IEC-14772-1:1997) by the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC) in December, 1997. There isn't space here to discuss the openness of the process, but ISO was so impressed by it that they're now studying it as a model for future standards development.

- **3D multi media:** Long before its official standardization VRML became the *de facto* standard for sharing and publishing data between CAD, animation, and 3D modelling programs; virtually every one of those programs now exports VRML or has a utility or

plug-in to convert its native file format to VRML. VRML is included or referenced in the upcoming MPEG-4 standard, Java3D, and in other developing standards.

- **Shared virtual worlds:** Being able to talk and work in a 3D shared virtual space was one of the earliest motivations of the VRML pioneers. The [VRMLworks] has a whole section on cyberspace that talks about the work that's being done to realize this vision.

- **On the Internet:** Unlike previous 3D applications, using the Internet to share 3D objects and scenes was built into VRML from the very beginning. The standard is even published in HTML.

4.4.1 What's the difference between VRML 1.0 and VRML 97?

Briefly, VRML 1.0 worlds are static and cannot directly interact with the user (they need a separate object tree to support interaction). VRML 2.0 worlds can move and interact with the visitor to those worlds. Here's a summary:

Version	Features
VRML 1.0	o Standard objects (cube, sphere, cone, cylinder, text) o Arbitrary objects (surfaces, line-sets, point-sets) o Ability to fly through, walk through, examine scenes o Lights o Cameras (viewpoints) o Textures on objects o Clickable links o Define and reuse objects
VRML 2.0	*All VRML 1.0 features plus* o Animated objects o Switches o Sensors o Scripts (Java or JavaScript) for behaviours o Interpolators (colour, position, orientation, etc.) o Extrusions o Background colours and textures o Sound (.wav and MIDI) o Animated textures o Event routing o Define and reuse objects and behaviours and effectively add new nodes to the language with PROTO and EXTERNPROTO

Another very important difference is that VRML 97 is an international spec approved by the International Organization for Standards: **ISO/IEC-14772-1:1997** and VRML 1.0 isn't.

4.4.2 VRML tools

In [tools] you will find an overview of most existing VRML generating tools. An example is [forms] a tool that creates a VRML image form a data set presented in a form.

4.5 X3D

4.5.1 What is X3D?

X3D is the next generation of the VRML specification. Like VRML it is a scene description language in a text file format. The major difference is that its syntax is XML rather than the older OpenInventor style syntax of VRML 1.0/2.0/97.

There is a loader available for the X3D format. See the Loaders section of the J3D.org site. This loader also is capable of loading the majority of the VRML 97 specification too.

X3D is designed to enable a new generation of XML-compliant 3D standards for the web. X3D stands for Extensible 3D. It is a next-generation, extensible, 3D graphics specification that extends the capabilities of VRML 97. The name X3D was chosen to indicate the integration of XML. As its name and circumstance suggest, the chief distinguishing factor of this technology is that it is an extensible architecture- meaning that *the processing of documents and 3D elements for display can take place under various "profiles" (software components) depending on the level and domain of functionality needed*.

This realization has a number of positive implications beginning with the fact that such a scheme allows one to streamline the minimum functionality needed for the viewing of 3D on the web and in web pages. In X3D, this minimum functionality is termed "The Core Profile". Necessarily following such a design is a re-assessment of the ontology of software components needed to view and experience various 3D content. With these additional profiles media designers can leverage the full potential of X3D's extensible architecture. For example, if one wanted to serve web-visitors VRML 97 content, the user would download the "VRML 97 Profile"; if they wanted to serve content with Multi-texturing or Multi-tracked audio or NURBS, the user would be required to download the appropriate profile. The ultimate benefit is that there is no monolithic plug-in that is expected to do it all- a familiar impediment for VRML developers and Working Groups looking to implement their improvements to the spec. Additional functionality can be designed as another profile and plugged right into the open X3D architecture; and the user only employs the profiles needed to view the specific content.

The Web3D Consortium has clearly defined a set of goals for X3D technology and is producing tangible results in their development of this technology.

- That its file syntax be compatible with the emerging World Wide Web Consortium's XML specification thus ensuring the tight integration of 3D into next-generation web browsers
- That it is backwards compatible and able to run all existing VRML content
- That its Core Profile is lightweight, able to be extended to provide new functionality
- That it is Extensible and able to use components (profiles) to add fundamentally new nodes and corresponding implementation code to the core

When we look at what's in the marketplace: many little technologies and little companies each trying to push their new 3D products. Those companies always have a few things in common: one is they solve a small piece of the puzzle very well; two, they have gotten a large infusion of cash from investors for this great idea, and three, they need to make a name for themselves. While this going on, VRML was the only file format trying to be interchangeable.

What we are doing is two-fold: we have structured X3D to scale in the direction to support all the small companies. We are trying to break out of that mode of competition and say "you all have good pieces, but they are small pieces, and no one wants to do all of it. So use X3D- it is a cross-platform, extensible technology." Then these smaller, excellent pieces have a place to land· and then these companies are able to show off their technology to best advantage (added to all the other interoperable pieces). So engage together rather than senselessly competing- *A rising tide lifts all the boats·*

So, X3D is not trying to be the final resting place-it is an extensible, an open ended architecture that will allow these things to co-exist and build off each other.

4.5.2 What does VRML in XML exactly mean?

Essentially, XML is just another file syntax. VRML 97 is one encoding for representing 3-D, a binary format could be another, and XML is another too. A trivial example showing one possible mapping between VRML 97 and XML might be something like:

```
VRML 97 : DEF MyView viewpoint { position 0 0 10 }

XML : <viewpoint DEF='MyView' position='0 0 10' >
```

Taking a slightly more complex example, the following illustrates how PROTOs could be defined and used using XML syntax. This example is taken from the latest version of the X3D DTD.

```
<Proto type="myBox">
  <Field id="mySize" type="vec">
  <Box size="mySize">
</Proto>
```

then somewhere else in a document

```
<ProtoUse type="myBox" DEF="aTwoThreeTenBox">
  mySize="2 3 10"
</ProtoUse>
```

and thereafter

```
<ProtoUse USE="aTwoThreeTenBox"/>
```

4.5.3 Is there an Open Source X3D browser?

Yes. The Java3D Working Group is producing an Open Source implementation of X3D called Xj3D. A working prototype is available today and has been tested under Linux, Solaris, and soon Win32. Current status is that a VRML 97 compliant DTD is available with a working X3D prototype example. In addition, Blaxxun has released their Contact VRML 97 browser as Open Source and intends to use this for their X3D implementation.

There are other open source VRML97 efforts that could potentially be used as the basis for an X3D browser, such as the work being performed as part of the OVAL project.

Finally, if the Cosmo Player source code becomes available in the future, then this could be used as another base for developing an Open Source X3D browser.

X3D is open source development with the interesting feature that it runs in the browser and consequently may unload the server if a lot of users concurrently access the VR scene.

It is also possible to use a XSLT stylesheet to transform X3D on the server directly to VRML97 and to visualize the 3D information directly with a standard Webbrowser with a VRML plugin. In that case no additional software is needed on the client.

[XML/X3D] gives an overview of X3D related tools available today.

4.6 Java3D

4.6.1 What is Java 3D?

Java3D is a low level 3D scene-graph based graphics programming API for the java language. It does not form part of the core APIs required by the Java specification. The class libraries exist under the javax.media.j3d top-level package as well as utility classes provided in javax.vecmath.

A low level API provides routines for creation of 3D geometries in a scene-graph structure that is independent of the underlying hardware implementation for real-time programming. The API provides scene-graph compilation and other optimisation techniques. It is heavily optimised towards the requirements of real-time 3D rendering and hence does not contain capabilities for photo-realistic rendering effects used to produce movie quality images (ie ray-tracing or radiosity based rendering algorithms).

The Java3D API consists of two parts: The API Specification and the implementation. Java3D mainly consists of the API specification. Anyone may implement the spec. Sun also provide an implementation of this specification but is encouraging 3rd party developers to implement J3D directly to the hardware.

4.6.2 What's the difference between Java 3D and OpenGL/Direct3D/PHIGS/, etc?

Java3D is another 3D programming API that exists on a similar level to OpenGL, Direct3D, PHIGS and similar systems. It is designed to use hardware acceleration wherever possible based on the underlying graphics architecture of the OS. That is, J3D provides a 3D rendering API for the Java language, but at the same time it may use OpenGL to do the interface to the hardware. J3D does not require direct hardware device driver support like the other APIs because it could rely on them to build its functionality. Sun is encouraging graphics board vendors to provide native implementations of J3D, but nothing has been seen yet.

4.6.3 What's the difference between Java 3D and VRML?

Java3D is a low-level programming API for 3D graphics rendering. The code must be compiled to move it to executable form. VRML is a text based modelling language that is interpreted dynamically from the source files.

The VRML Consortium and Sun are working together to implement a Java3D-based VRML browser. In September 1998 Sun released the source code for unrestricted use. A working group is now developing the code. The official beta 2 release and the working group can be found at

http://www.vrml.org/WorkingGroups/vrml-java3d/

4.7 Scalable Vector Graphics (SVG)

4.7.1 What is SVG?

SVG is a XML-language for two-dimensional scalable vector graphics. SVG allows for three types of graphic objects: *vector graphic shapes* (e.g., paths consisting of straight lines and curves), *images* and *text*. Graphical objects can be grouped, styled, transformed and composited into previously rendered objects. Text can be in any XML namespace suitable to the appplication, which enhances searchability and accessibility of the SVG graphics. The feature set includes nested transformations, clipping paths, alpha masks, filter effects, template objects and extensibility.

SVG drawings can be generated from bcXML content besides HTML and are dynamic and interactive. The Document Object Model (DOM) for SVG, which includes the full XML DOM, allows for straightforward and efficient vector graphics animation via scripting. A rich set of event handlers such as onmouseover and onclick can be assigned to any SVG graphical object. Because of its compatibility and leveraging of other Web standards, features like scripting can be done on SVG elements and other XML elements from different namespaces simultaneously within the same Web page.

SVG supports 2D vector drawings with text that can be made subject to a Context parameter that steers the language and the user type.

4.7.2 Do not forget paper

SVG can have several interesting applications that will be explored in WP6 and nicely bridges the gap between the paper world of the past and the electronic world of the future. Paper as a means to communicate will exist for many decades, ignoring that fact is not useful. Interesting is the application of SVG (in combination with XSL formatting objects) as an electronic paper format that can be steered by the same XSLT style-sheet that also produces HTML and X3D. One source several targets. With as added value the chance to maintain one version of the data and produce printable output at will.

Because SVG is an XML grammar, SVG graphics can easily be generated on web servers "on the fly," using standard XML tools, like XSLT. For example, a web server can generate a high quality, low bandwidth stock quote graph from stock market data. Directly from the stock data flowing into the system.

Or, it can generate the drawing of a building element by dynamically translating the dimensions and geometric properties of the building element into a visual representation, with reference all dimensions to the drawing.

[SVG]

4.7.3 Example

The rectangle example below expresses all values in physical units (centimeters, in this case). The rect element is filled with yellow and stroked with navy. This code is from the W3C working draft. (Note: Lines of code are broken for display purposes only.)

```
<?xml version="1.0" standalone="no"?>

<!DOCTYPE svg PUBLIC "-//W3C//
DTD SVG 20000303 Stylable//EN"
"http://www.w3.org/TR/
2000/03/WD-SVG-20000303/
DTD/svg-20000303-stylable.dtd">
```

```
<svg width="12cm" height="4cm">

<desc>Example rect01 - rectangle expressed in physical units</desc>

<rect x="4cm" y="1cm" width="4cm" height="2cm"
style="fill:yellow; stroke:navy; stroke-width:0.1cm" />

</svg>
```

4.7.4 Pros and Cons

SVG has many advantages over other image formats, and particularly over JPEG and GIF, the most common graphic formats used on the Web today. Specifically:

- **Plain text format** -- SVG files can be read and modified by a range of tools, and are usually much smaller and more compressible than comparable JPEG or GIF images.
- **Scalable** -- Unlike bitmapped GIF and JPEG formats, SVG is a vector format, which means SVG images can be printed with high quality at any resolution, without the "staircase" effects you see when printing bitmapped images.
- **Zoomable** -- You can zoom in on any portion of an SVG image and not see any degradation.
- **Searchable and selectable text** -- Unlike in bitmapped images, text in SVG text is selectable and searchable. For example, you can search for specific text strings, like city names in a map.
- **Scripting and animation** -- SVG enables dynamic and interactive graphics far more sophisticated than bitmapped or even Flash™ images.
- **Works with Java[tm] technology** -- SVG complements Java[tm] technologies' high end graphics engine, the Java 2D API.
- **Open standard** -- SVG is an open recommendation developed by a cross-industry consortium. Unlike some other graphics formats, SVG is not proprietary.
- **True XML** -- As an XML grammar, SVG offers all the advantages of XML:
    - Interoperability
    - Internationalization (Unicode support)
    - Wide tool support
    - Easy manipulation through standard APIs, such as the Document Object Model (DOM) API
    - Easy transformation through XML Stylesheet Language Transformation (XSLT).

4.8 AutoCad

Also in the 'traditional' electronic drawing world of AutoCad there are developments that in some cases may be relevant, i.e. generating VRML from DXF/DWG, and supporting the upgrading from the points and lines approach to a product modelling approach (Architectural Desktop). Because these developments are only curing a problem that does not exist if the designer is using a 3D modeller instead of the outdated points and lines modeller, we leave this technology for what it is. See also below in 4.7 and 4.8.

4.9 Geometric Modellers

Generation of VRML from a geometrical modeller is one of the possibilities that are useful for the of-the-shelve and to a lesser extend parametric object classes. Especially if the shapes are complicated, like say for the category door knobs, VRML generation from bcXML files is not possible.

All the main geometric modellers currently are able to export VRML.

4.9.1 Geometry Exchange standards

Another possibility to produce and communicate shapes of construction objects is to communicate by using a standard geometry exchange format like STEP AP 42 and to generate a VRML from such descriptions.

Though this might be true it is a low priority option as STEP is not very influential in our sector of industry.

4.10 Product Modellers

Another possibility is to generate VRML from a product modeller - in our case a building modeller - like: ArchiCad, Architectural Desktop (ADT), Allplan, and such. The difference between a product modeller and a geometric modeller is the way they treat semantics. In a geometrical modeller, shapes can only be understood by the users.

In a building modeller columns and beams are known internally as columns and beams with certain typical column and beam properties. This data is called *definition* data. Different shapes can be attached to instances of these classes. In fact the same image as produced by a geometric modeller can be used here to. In most cases however a shape representation of an object will be generated on the fly from the definition data.

It is true that sometimes geometrical modellers support some labelling technique that shows meaning of the object in words, but that does not mean that the systems 'knows' what the object is.

Most current product and building modellers are able to generate VRML from their internal representation.

4.10.1 Product Model Exchange Standards

In BC the IAI-IFC is one of the most influential standards. Also it is possible to generate VRML from the neutral IFC shape description. Several institutions have even developed VRML based IFC browsers [TNOifc, VTTifc].

The application of this type of technology will undoubtedly play a role in the design/engineering area.

Figure 1 below shows the typical use of the IFC-browser. At the left you see the object tree of the building displayed. Elements of the tree can be selected and manipulated.

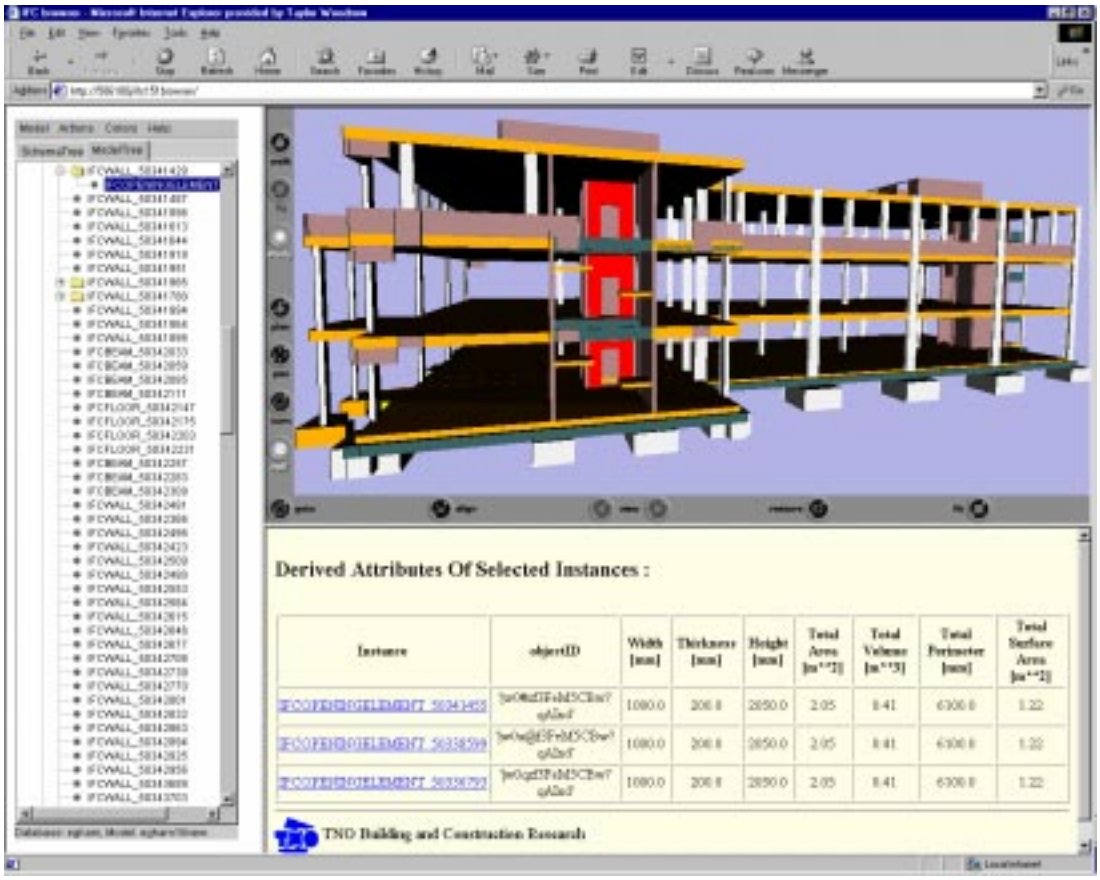


Figure 1: IFC browser screen dump

Figure 2 below shows in detail how a simpler object, in this case a stair, is treated in the IFC. It shows the definition of a stair, the geometric representation, an illustration and the formal Express schema.

IfcStair

Definition from IAI: A vertical passageway allowing occupants to walk (step) from one floor level to another floor level at a different elevation. It may include a landing as an intermediate floor slab.

The stair is a container entity that aggregates all components of the stair, it represents. The aggregation is handled via the *IfcRelAggregates* relationship, relating a stair (*IfcStair*) with the related flights (*IfcStairFlight*) and landings (*IfcSlab* with type 'Landing').

Geometry Use Definitions:

The geometric representation of *IfcStair* is given by the *IfcProductDefinitionShape*, allowing multiple geometric representations. Independent geometric representations should only be

used when the IfcStair is not defined as an aggregate. If defined as an aggregate, the geometric representation is the sum of the representation of the components within the aggregate.

Local placement

The local placement for IfcStair is defined in its supertype IfcProduct. It is defined by the IfcLocalPlacement, which defines the local coordinate system that is referenced by all geometric representations.

The PlacementRelTo relationship of IfcLocalPlacement shall point (if given) to the local placement of the same IfcSpatialStructureElement , which is used in the ContainedInStructure inverse attribute, or to a spatial structure element at a higher level, referenced by that.

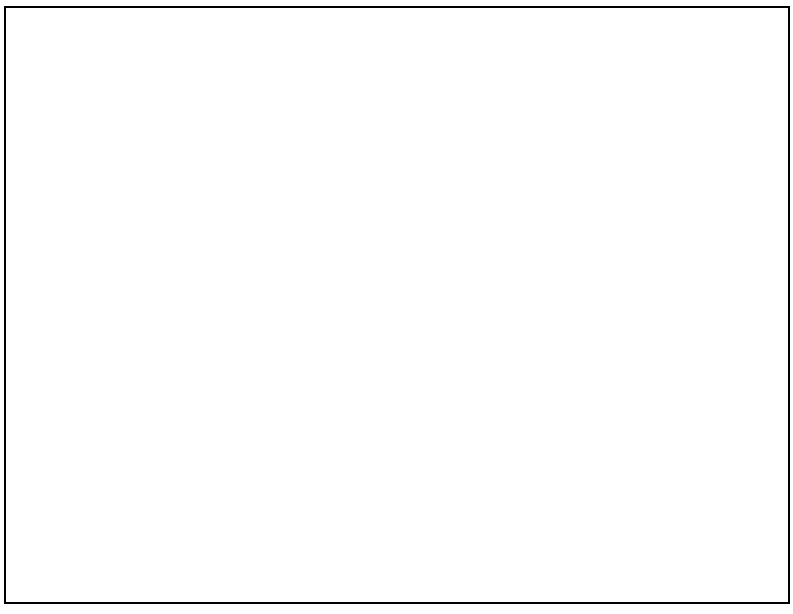
If the relative placement is not used, the absolute placement is defined within the world coordinate system.

If the LocalPlacement is given for the IfcStair, then all components, which are aggregated to the stair should use this placement as their relative placement.

Geometric Representation

If the IfcStair has components then no independent geometric representation shall defined for the IfcStair. The IfcStair is then geometrically represented by the geometric representation of its components. If the IfcStair has no components defined then the IfcStair may be represented by an IfcShapeRepresentation with the RepresentationType = 'Brep'.

Illustration:



IfcStair defining only the local placement for all components.

EXPRESS specification:

```
ENTITY IfcStair
  SUBTYPE OF (IfcBuildingElement);
    ShapeType            : IfcStairTypeEnum;
  WHERE
    WR1       : (HIINDEX(SELF\IfcObject.IsDecomposedBy)     =     0)     OR
                ((HIINDEX(SELF\IfcObject.IsDecomposedBy)     =     1)     AND
```

```
                ((HIINDEX(SELF\IfcObject.IsDecomposedBy)    =    1)    AND
                (NOT(EXISTS(SELF\IfcProduct.Representation))));
  END_ENTITY;
```

Attribute definitions:

ShapeType : Predefined shape types for a stair that are specified in an Enum.

Formal Propositions:

WR1 : Either the stair is not decomposed into its flights and landings (the stair can have independent geometry), or the geometry shall not be given at IfcStair directly.

References (1):

Name	Type	Referred through	Express-G
IfcBuildingElement	Entity	Subtype	Diagram 1

Inheritance graph

```
  ENTITY IfcStair;
    ENTITY IfcRoot;
  END_ENTITY;
```

```
END_ENTITY;
```

Figure 2: Example: ifcStair

What the example shows is the use of Express and the ifc as a formal means of producing a tightly coupled definition of an object and its shape representation. It also shows that this approach can be used for a class of objects that play a role in eCommerce.

4.10.2 STEP and XML

Of course also ISO-STEP is concentrating on the relation between Express and XML. ISO-STEP is currently defining standard ways of (1) mapping STEP Physical Files to XML and (2) mapping EXPRESS to DTD/XML Schema. The IAI is harmonizing with STEP and putting a lot of effort in transforming STEP based information (both data and structure) in the XML world. The generation (automated by a Perl script) of XML Schema generation from EXPRESS is discussed in close co-operation with eConstruct. We therefore are in the position to exactly comply to the XSD generation fixed by the IAI (and used by them to map ifc2x to ifcXML). This ifcXML will, beyond a Common Object Schema (COS) that is re-used in bcXML, also potentially provide their explicit shape information in XML form, which could be used as shape definition from which an actual presentation can be derived.

[ifcXML document]

4.11 IfcXML

eConstruct is not the only group working on XML based communication in BC. One other development is the work of the IAI on ifcXML.

4.12 Shape Grammars

Shape grammars are mathematical languages used mainly in Architecture for designing the spatial outline of a building. Most often they can generate VRML. For eCommerce this class is not relevant.

4.13 Parametric Shape Languages

Another class of possibilities is the use of parametric shape languages. There are two types of parametric shape languages. The first purely extends the capabilities of geometric modellers. An example is STEP Part 108. The second takes a broader vision and focuses on parametric objects as found in BC, like for example a row of similar lampposts, or a pattern of similar windows.

Mainly dedicated to the needs of the Mechanical industry ISO 13584 P-lib is of interest.

[AP221/Epistle]

4.13.1 Geometric Description Language (GDL)

One of the most interesting examples of an architectural shape language is Graphisoft's Geometric Description Language [GDL]. GDL is a parametric programming language used to

describe the shapes of doors, windows, stairs etc together with the 2D symbols representing them on a floor plan. These are called library parts. Every library part described with GDL has scripts. Shapes of different levels of detail can be attached to the same object.

In order to promote GDL a group of companies formed the GDL Alliance with the following mission statement: To promote the culture of GDL as a world-wide standard for creating parametric 3D objects for use in 3D models and virtual buildings and enhance the end user experience by providing an environment to attain a more clear understanding of GDL capabilities and implementation (http://www.gdlalliance.com/)

Although GDL started as a Graphisoft development it is now getting wider attention and users are no longer required to buy a ArchiCAD license. A free browser plug-in is available from: http://gdl.graphisoft.com/

In [Objects] you can see some examples of library parts.

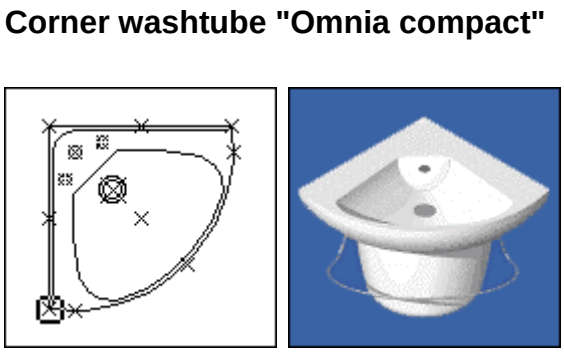


Figure 3: GDL Object

The script language that describes these objects in detail is quite simple and powerful.

4.14 ISO 13584 P-lib

This International Standard, prepared by ISO/TC184/SC4/WG2, uses the same modelling language (EXPRESS) and implementation methods as STEP (Physical File, SDAI, ...) and provides powerful capabilities, complementary to those already provided by STEP, for modelling and exchanging product model data.

STEP provides for EXPLICIT (and very complete) modelling of ONE Product. Parts Library provides for IMPLICIT (and possibly simplified) modelling of FAMILIES of cognate products.

Using the Object Oriented terminology, STEP enables modelling and exchanging INSTANCES of products, Parts Library provides for modelling and exchanging CLASSES of products, whether these products are catalogue parts (e.g., INA TKDS family of linear bearings) or whether they are logical components such as "elbow" (in piping) "resistor" in electronics, ...

The various parts of the ISO CD 13584 multi-part specification have already been prototyped on various Computer Aided Design and Engineering system environments (PC-based, workstation/mainframe-based, parametrics, object-oriented) and using various storage technology (OODB, RDB, ER-based DB, KDB) in several national or international projects.

Some of the key concepts of the ISO CD 13584 approach are already implemented in different commercial software products (e.g., concept of semantic dictionary in SPIMS from SGAO, concept of multi-representation in Cas.Cade from Matra Datavision, program-based parametrics specification in PRONOS/PRIAMOS from VW-Gedas, ...).

New projects are now launched both in Industrial area, and in the International Standardization area (e.g., in ISO TC 29 : "cutting tools") that use the information models specified in ISO CD 13584 to progressively build the computerized (and standardized) dictionary of product properties required for a mature product data technology.

On one hand, Parts Library provides capability to describe (and exchange) complete structured sets of catalogues data defined by thousands of Suppliers (Standardization Bodies, commercial Suppliers, Internal Company Specifications), but, on the other hand, it also provides resource constructs that should fulfil requirements that emerged from several SC4 projects.

These resource constructs includes the following capabilities:

- Model and exchange generic expressions and, in particular, all the numeric, Boolean and string expressions conforming to the EXPRESS Reference Manual (ISO CD 13584-20);
- Specify, and exchange, classes of cognate products and of product representations;
- Reference from product data, a component class or a representation class, whether this class is externally defined, or whether it is exchanged together with the product model data;
- Specify, at run time, product property definition and to exchange also values of these properties;
- Reference product property definitions that are externally defined;
- Associate constraints on product properties;
- Model and exchange tables.

4.15 TU-Delft Semantic Shape tool

Another interesting development is the Semantic Shape tool developed by the TU-Delft. The shape tool’s meta-model describes the world in 4 classes of objects classified as: Point-like, Line-like, Face-like and Volume-like objects. The classification according to the main dimension(s) of the object is artificial, but ideal for geometry and topology modelling.

These four object types are root-objects and all the BC-product related objects fall into one of these four classes. As these objects are still *semantic* objects (things that exist in the real world) and not only the shapes of semantic objects, they can be represented with different shapes (i.e. a Line-like object can be represented by a stretched face and look like a road).

Additionally the model defines a set of traditional relations between these objects of the type: consists-of, bounded-by and connected-to, and a set of non-manifold relations of the type: on, in, etc (point on line, face in face).

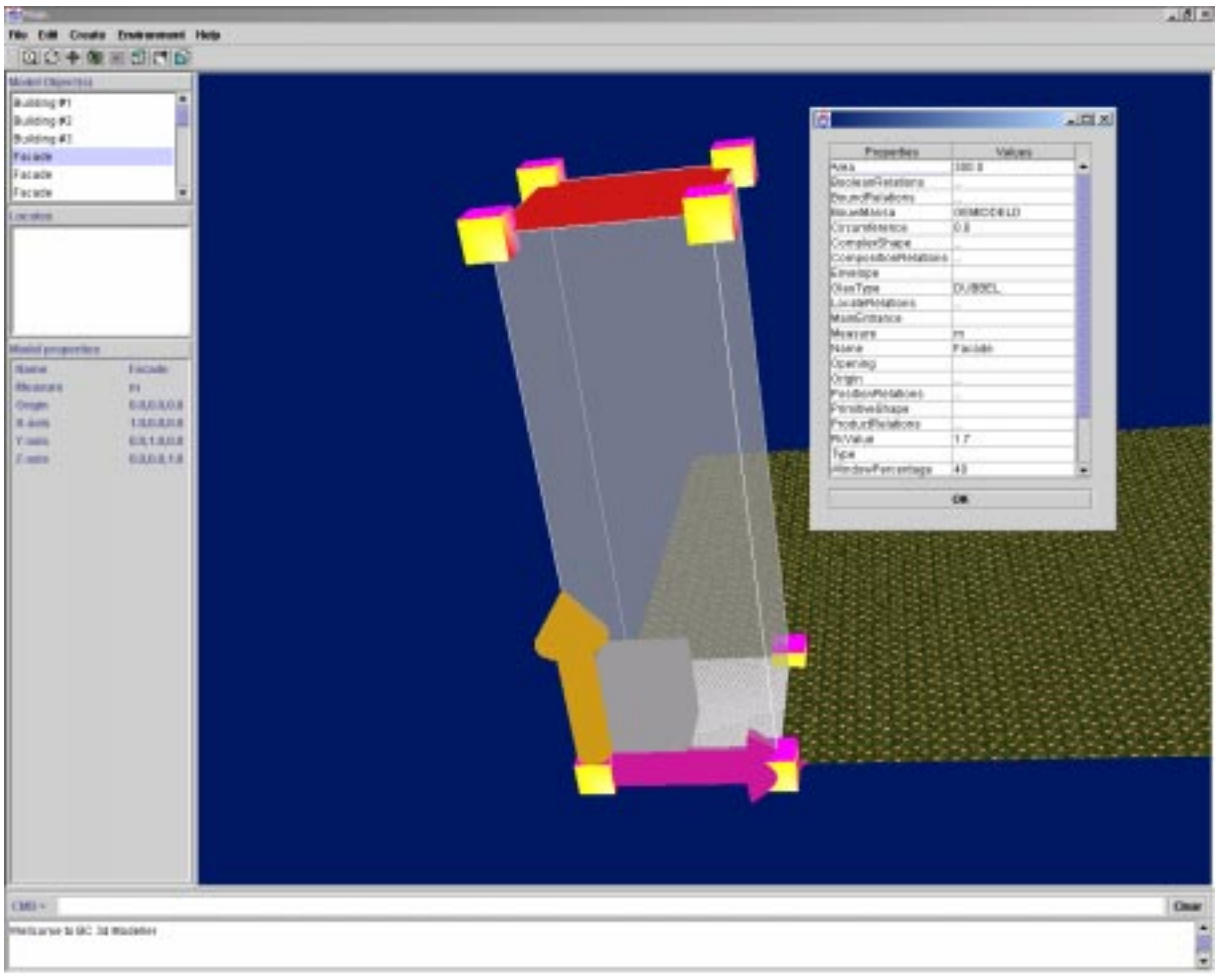


Figure 4: Impression of the TU-Delft Semantic Shape tool

The decomposition of each object reveals greater detail, which again can be formed by sets of semantic shape objects. Based on this model a Java3D tool has been developed that is used to produce a building model that can be shared and - to some extend - manipulated over the Internet. An example of an application is that end-users with the right authority can make simple changes in the model; they can change colours, change textures like tile designs, change locations of electrical outlets, etc.

4.16 Speech Processing

As bcXML provides true BC semantics an interesting new opportunity is the application of speech technology, another branch of multi-media technology. The simplest case is of course to add *.wav* sound elements to the BC Building Definitions. This could result both in speech output and in speech input (speech recognition) in different languages. Also speech control could play a role in the navigation process, i.e. say “left” instead of using the mouse. Another XML-based technology, VoiceXM, can be used to realize voice-controlled browsing and navigation over the telephone.

Though these technologies are clearly very interesting for the future, they are out of scope of the current research which mainly focuses on graphics.

4.17 Conclusions on Technology

The amount of shape and VR-related developments is overwhelming. What seems clear is that VRML is the most popular scene description format in use. With X3D the popularity of VRML will increase. Java3D is the obvious choice for building multi platform graphical applications.

For the future it is important that suppliers provide images of their library (standard) parts in VRML format. One approach is particularly interesting, i.e. the Graphisoft GDL approach that provides VRML generation in the browser. Other interesting approaches use STEP technology (IAI-IFC and ISO 13584), which in time will be XML-ified.

At this point in time it is becoming more and more clear what the values are of each individual technology for use in combination with bcXML. X3D and SVG are particular interesting because of the common XML basis with bcXML. XML supports the features required to implement distributed information sharing and the common use of the Internet for project support. Other missing components like increased security and digital signatures are all becoming available. Increased processing power and increasing communication bandwidth support 2D and 3D bcXML enhanced communication.

At the other hand, X3D is still in its infancy, with all the related problems. SVG has as additional advantage that it does not require the navigation of VR scenes and fits well with the traditional technical drawing approach used in the BC industry. Besides that, it is also suitable for use as a basis for traditional based paper output. This can be achieved by integrating SVG within XSL Formatting Objects and in a later stage render it to for example PDF.

Regarding the SVG and X3D output it is also possible to implement some level of interactive questioning and information hiding. This means that one can interactively question for more information by interacting with the 2D picture or 3D environment; this can result in clickable parts within the picture / environment which will produce more information about a specific part of the object. This interactive hiding of information will also result in the possibility to select different views on information. For instance the outer view of the object stating its dimension versus the inner view showing its internal construction like reinforcement beams. Also this change of visualisation will be possible by interacting with the picture real time.

Another question is: how to combine an IFC-world with the more detailed world of parts and components targeted in this research. This question will undoubtedly be targeted by the IFC-community.

In T6200 we will implement a subset of the technologies described above and compare their features relevant for our purpose.

5 APPLICATION OF VR IN BC

In this section we are matching the industry needs with the available technology, i.e. which needs/opportunities can benefit from a multimedia enhanced bcXML and which technology could play the role?

5.1 Procurement of Goods

In procurement VR can play a role because it is not always simple to find or configure suitable products if only supported by pictures from glossy folders. Even the simplest use cases can benefit from an information front-end as it is quite often necessary to obtain additional information about components or features that is not clear from the description provided, like the exact measurements in 3D. By using X3D and VRML it is possible to provide the end-user with the required functionality through his browser.

For more complex cases VR information front-ends are indispensable. They are the best interface possible, are intuitive and simple to navigate, and allow even the simplest user to find what he wants.

Looking at the three product classes: standard, parametric and custom, the situation seems to be as follows.

5.1.1 Standard

Only the vendor can model his standard library parts. For eConstruct we can simply assume that a VRML will be available. The VR front-end will then need functionality to: (1) combine VRMLs of different components, (2) make the object and its parts sensitive, (3) produce a pop up window with information in different languages if requested.

5.1.2 Configurable

For the class of configurable objects things are somewhat more complicated. The most interesting approach should realise a combination between the parametric shape description and bcXML. Changing the bcXML data immediately operates on the script and subsequently on the VRML. How this can be done is one of the research subjects currently being investigated.

5.1.3 Custom

For the made-to-order objects the situation is again different. The main reason is that much more communication between client and supplier is required. If the supplier still is a human, it might be an idea to provide the consumer with the functionality to generate and communicate FU-shape (abstracted shape description only to illustrate the request). If the supplier is a modeller the problem is much more complicated because of the uncertainty of the meaning of various parameters.

Communication about custom design objects can be supported by VR only if it can be communicated by bcXML, which is probably not possible in the general case.

5.2 Procurement of Specialist Services

One problem that still has to be solved is how to relate bcXML to the ICT developments that are taking place in sectors related to BC like Mechanical and Electrical? As an example consider the work done in the ISO 13584 Parts Library project for Mechanical Engineering, or STEP 221 for the Process Plant industry. Though it seems possible that all these ICT developments will produce XML versions of their standard, and – in theory – it should be possible to use bcXML with the taxonomies developed in those fields, nothing is certain unless its proved, which unfortunately is not possible to do within the scope of the current project.

What could be done however is to design and develop bcXML so that it will be possible to interface external taxonomies and to communicate *over* different taxonomies.

5.3 C2B/B2C

This use case scenario [tolman] (besides 6.1) is the one that WP6 is particularly focusing on. In fact it is the same as above, but with the additional restriction that users are not experienced and have no other tools (like IFC-modellers).

One of the possibilities that may be of interest is that even civilians that don't speak a foreign language will be able to find information about components abroad and read the specs.

5.4 H2H

Internet based communication between humans on a large-scale is the ultimate goal of VR information front-ends. On a smaller scale the technology can be applied in case of custom designs where a customer wants to express his needs with the help of visual aids. Provided that the bcXML definition includes decomposition (and a description of its parts) FU-shapes can easily be generated from bcXML input data. For example figure 2 shows a FU-shape of a table with some simple properties: height, width, length.

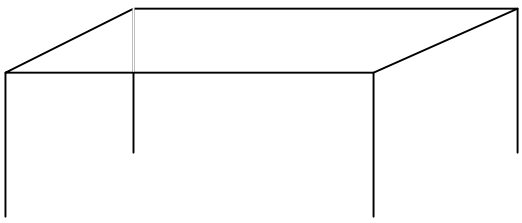


Figure 5: FU-shape of a table

An interesting application of FU-shapes can be in the end-user GUI. If a user defines a configurable object like a stair or a dormer it might be useful to show him how the system understands his request and show him a FU-shape image. This feature may well reduce the number of erroneous requests where the user input is not following the definitions used in the system. Another example of FU-shape is presented in Figure 2, which shows an image of a stair; not a TS-stair as a vendor would produce it.

5.5 Design/Engineering

In the design/engineering stage the most promising development of today is the IAI-IFC. eConstruct was therefore very fortunate to co-operate with the IAI in the development of a Common Object Schema (COS) shared between bcXML and ifcXML.

As the IFC format already includes a neutral shape definition that can be translated into VRML (see figure 1) and it is also no longer mandatory to own an IFC modeller, the ifc browser can be used without a modeller, in this stage VR is already well established.

As also mentioned in 4.10 the relation between IFC and bcXML will be in focus in T6200.

5.6 Specification/Estimation

A VR information front-end also can play an important role in the specification/estimation stage of a project as it might support the users with up to date information about products and services. One improvement upon the current process might for example be that it may be simpler to show which measurements are important and how the measurements should be taken. Another interesting feature might be the close relation between the VR and the actual library parts.

5.7 Realization

In the realization stage there are more multi-media aspects coming into scope. An example is to add movement and to support simple simulation. This might be interesting for supporting assembly steps. Another aspect might be the addition of the time dimension (4D time scrolling). Yet another case will involve the addition of sound to support the user that wants to know what he will hear from the construction work, or the traffic noise.

Much works has already been done on VR support for the realization stage of construction projects [Edwin Dado, Opdenbos].

Figure 6: Excavator

5.7.1 Mobile interface “m”

One particular interesting opportunity is the connection with mobile applications and speech processing technology. As bcXML provides the right semantics including the translations in several languages the next step in this direction might be to add the pronunciations. If available it becomes possible to interact with an application or equipment that uses voice input and output. This might serve the needs of, say, supervisors or the step-wise instruction of assembly or maintenance workers. Especially on-site applications that withstand dirty weather conditions seem interesting.

5.8 Maintenance

Also in the maintenance stage there are opportunities for VR support. However most applications have been covered already in the above. One additional observation is that after a couple of years it is often impossible to replace a defected object with exactly the same object as they are no longer for sale. It therefore might be of interest to be able search objects with certain fixed geometrical constraints (as to make them fit into place).

6 SCOPE FOR T6200

As seen above, matching the industry requirements (or wishes) with the technology gives a large set of opportunities for software development, which far outreach the possibilities of task 6200.

We therefore decided to focus on what we think are the core problems and core technologies and apply them in a restricted domain of BC, i.e. *eCommerce of configurable objects*.

The core technologies are seen to be SVG and X3D and some parametric shape language or parametric language constructs in bcXML, as they fit best in the chosen C2B / B2C use case scenario and form the basis of nearly everything else related to VR information front-ends. Extensibility makes lightweight solutions possible (tailored to the shapes in the domain). The common XML-basis simplifies the connection with bcXML.

6.1.1 bcXML Reference Architecture

The work on multi-media extension and presentation fits well in the bcXML Reference Architecture currently under development [van Rees].

With bcXML information and a multi-media formatting file it will be possible to apply XSLT transformations that achieve a 2D (SVG) and 3D (X3D) output view of the input bcXML data. Besides this, also different textual formatting outputs will be covered.

The WP6 software development efforts will be delivered in two different ways:

- At first it will consist of a stand-alone application, which will generate textual, 2D and 3D output based on a bcXML data file and a multimedia formatting file.
- Besides this stand-alone application, WP6 will support the development of the bcXML Reference Architecture by taking care of the design and implementation of the data visualization component of within the architecture. This is the reason that the software, we will call “the Visualiser” will consist of more than the capabilities to render 2D or 3D images solely. Besides this it will also take care of general textual output based on bcXML data and taxonomy information.

The Visualiser will be integrated at the presentation part of the Reference Architecture. The main reason to extend the Visualiser with textual outputting capabilities - and with that extension covering textual formatting needs of the Resource Browser in the Reference Architecture - is that the technology needed here is in line with a subset of technologies needed for the generation 2D and 3D visualizations. This way we can prevent ourselves from doing double work.

7 OVERALL CONCLUSIONS

D601 presents the results of a study into multi-media enhancements of bcXML; what are the possibilities, what are the applications, what are the benefits?

As to the industry needs (the opportunities) the main conclusions are that a wide range of multi-media and especially 2D (SVG) and 3D (X3D) enhanced bcXML applications can be identified, and that VR as an information front-end seems quite useful as it will eliminate a lot of errors. The bcXML semantics and the multi-lingual facility of bcXML really facilitates co-operation between partners of different member states and a VR front-end can apply this feature to present information in the language of choice. Moreover, as everybody nowadays has access to the Internet, early involvement of authorities, suppliers and public can be made possible and may well facilitate improved project execution and participation processes.

Graphics support for eCommerce seems most beneficial in situations where configurable shape descriptions can be closely coupled with bcXML (figure 7), and where information hiding is required, i.e. don't show what is not immediately relevant (look at the hinges and door knobs of figure 7). Also the co-operative design of configurable objects that involve design input from several parties can greatly benefit from the results of WP6, like say the design of reinforced concrete slabs with openings and holes for stairs, ducts, service-pipes and cables (see figure 8), as the technology developed is cheap and generally available.

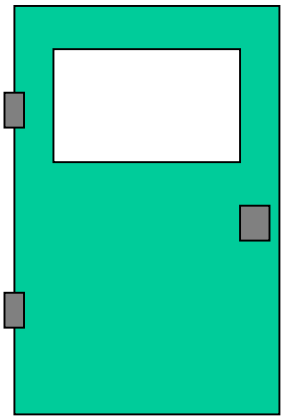


Figure 7: 2D impression of a panel door with hinges and glass opening

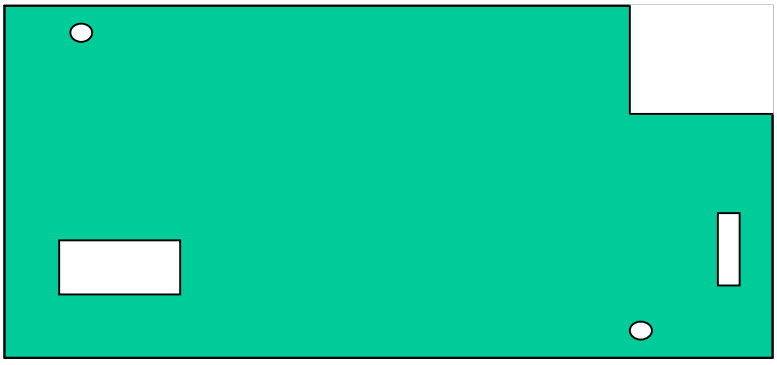


Figure 8: 2D impression of a floor slab with openings and holes

The 2D and 3D presentations provided can be augmented with textures and texts, as shown in figure 9.

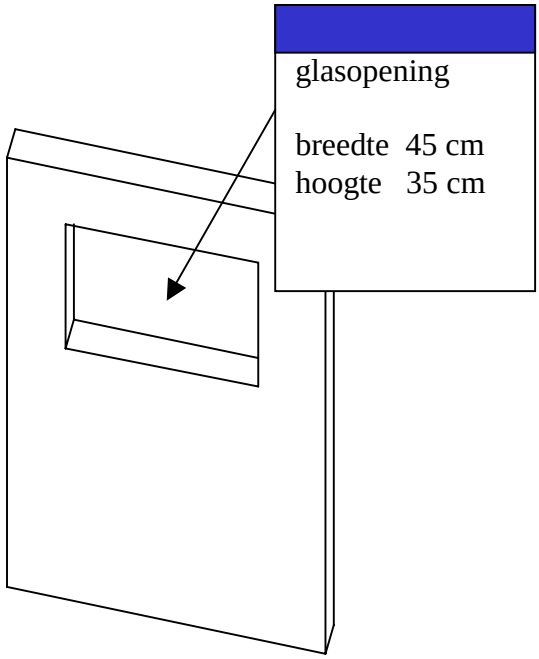


Figure 9: 3D impression of a door with pop up window (language = Dutch)

Upon clicking the glass opening in figure 9 a pop-up window shows details in the requested language and alphabet. Information about selected details can be provided in text using HTML, or again in 2D (SVG) and 3D (X3D). Finally, using ebXML technology, a view mechanism can be implemented that controls the presentations given the role and/or the authorisation of the user.

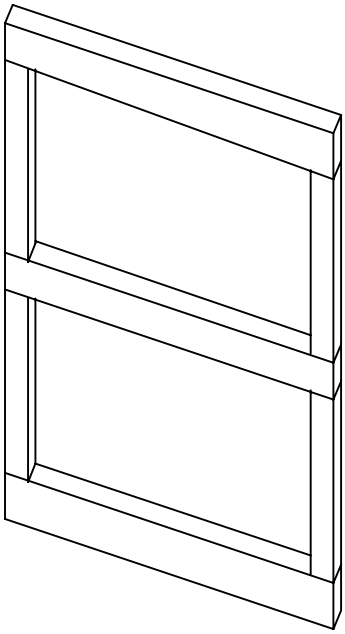


Figure 10: 3D impression of the frame of a panel door (view = *inside*)

Additionally - though out of scope of WP6 – bcXML makes it possible to use speech control (input and output) to help the user in browsing the VR scene.

For standard products VR enhancement is quite simple and totally for the suppliers responsibility. There are several commercial CAD-systems available that can generate VRML from their internal format.

In BC custom design Product Data Technology is increasingly becoming state-of-the-art. In those cases the IAI-IFC and IFC supporting product modellers and IFC-browsers come into view. The COS shared between eConstruct and the IAI provides a helpful bridge.

As to the overall objectives of the project and of WP6 in particular the main conclusions are:

- the eConstruct technology is cheap but powerful and suites the needs of the European Building and Construction industry
- bcXML potentially provides the semantics required by the BC industry
- the bcXML neutral object definitions can be used to generate 2D and 3D presentations from the same content as used in HTML formatting
- the bcXML multi language facility greatly improves communications over the national borders also if used to augment 2D and 3D graphical presentations
- the bcXML view mechanism can be used to present the user with the right information only
- the fact that the 2D and 3D presentations can be used to find the required information and – to some extend – can also be used to input new data, clearly illustrates the strength of 2D and 3D graphics as an information front-end.

How the functionality explained in this report is actually implemented will be described in the deliverables related to task 6200.

8 REFERENCES

[VRMLworks] http://home.hiwaay.net/~crispen/vrml/build_intro.html

[tools] http://home.hiwaay.net/~crispen/vrml/build_intro.html

[forms] http://www.ncb.mb.ca/vrml/

http://www.virtuworlds.com/3DEZine/art_x3d.html
http://www.web3d.org/TaskGroups/x3d/faq/index.html

[tolman] *Building and Construction eXtensible Mark-up Language (bcXML): the C2B / B2C scenario*, paper CIB 2001

[SVG] http://www.xml.com/pub/a/2000/03/22/style/index.html

http://www.w3.org/Graphics/SVG/Overview.htm8
http://www.w3.org/TR/2000/CR-SVG-20000802/interact.html
http://www.webdeveloper.com/design/design_svg_intro.html
http://www.sun.com/software/xml/developers/svg/
http://www.oreillynet.com/pub/a/network/2000/04/28/feature/svg.html

[XML/X3D] http://www.web3d.org/TaskGroups/x3d/translation/XmlToolRelationshipsForX3d.pdf

[TNOifc]

[ifcDocument] www.eConstruct.org

[VTTifc]

[GDL] http://www.gdlalliance.com/

[Objects] http://www.archimedia.de/GDL_was.html

[Edwin Dado] *Improving Electronic Communication on the Construction Site of Large-Scale Projects, PhD thesis, TU-Delft, 2001*

[van Rees at all] *eBusiness in Building Construction: the eConstruct Project, e2001 conference, October 2001, Venice, Italy*

AP221/Epistle http://www.stepcom.ncl.ac.uk/epistle/ap221/dwnld221.htm