A Network-based Kit-of-parts Virtual Building System

A. SCOTT HOWE, ARCHITECT
Kajima Corporation and The University of Michigan
School of Architecture and Urban Planning
email: ash@ipc.kajima.co.jp

Abstract

This paper describes an experimental browser / modeler which will allow the user to collect and assemble virtual kit-of-parts components from “component libraries” located on the Internet (such as manufacturer’s databases) and assemble them into a virtual representation of a building. The fully assembled virtual building will provide a basis for ordering and manufacturing actual components and preparing for construction. The browser will allow the designer to affect a limited degree of remote fabrication at real manufacturing facilities, and facilitate eventual interface with built in sensors and actuators. The browser will manipulate and display interactive three dimensional objects using Virtual Reality Modeling Language (VRML). Upon assembly, actual components will have sensors built into them for providing data about the real building, which could be viewed during a walkthru of the virtual building by clicking on parts of the model. The virtual building will work as a remote facility management tool for monitoring or controlling various architectural devices attached to the real building (such as electrically driven louvers, HVAC systems, appliances, etcetera).

1. A Research Programme

Increasingly, powerful computer-aided design tools have enjoyed greater roles in the design process. In parallel, the Internet with its World Wide Web has proved to be a revolution in information dissemination, providing real-time access to sources located around the world. Since information is the ultimate substance from which designs are conceived, a logical question could be: how can a design process be enhanced by direct links to information sources? Considering the increased proliferation of information-based automated manufacturing processes, a second question would logically follow the first: how would direct instantaneous access to manufacturing processes affect the design process? Finally, instantly gleaning performance data from the constructed object itself, how would an information feedback loop affect current use and future design improvements to future objects of a similar type? While these questions will probably remain unanswered for many years, the development of an experimental environment which sets the stage for linking design, manufacturing, and use can be facilitated.

As a research programme, it is proposed that a computer tool be developed which can affect such an experimental environment. The computer tool has been conceived in the form of a plug-in to a common Internet browser. Some functions of the plug-in will eventually be made available to a selected number of designers for further research purposes.

2. VBuild Browser

2.1 VBUILD CONCEPT

The browser plug-in (hereafter called VBuild) makes maximum use of two powerful concepts: object-oriented programming and kit-of-parts philosophy. In a way, the two concepts work hand in hand.

2.1.1 Object Oriented Programming

VBuild was programmed using an object-oriented structure in the C++ language. Object-oriented languages utilize data and code in discrete structures called classes. Once instantiated, the class defines types of data called objects. The data itself is protected and can only be manipulated via pre-defined methods. The methods are interfaces with the rest of the program. Once the interfaces are defined, the actual coding for implementing the interfaces can take on any form as long as it supports the interface methods. In this way specific elements in a model can be defined independently of other code according to function or behavior. Methods and attributes specific to that element's behavior can be added as needed to give the coding completely expandable capabilities.

In an object-oriented program, once a class has been debugged it becomes a reliable building block in the entire coding of the program. Object-oriented design becomes a clean way of defining functionality without having to deal with loose ends associated with the complexities of unstructured raw coding.

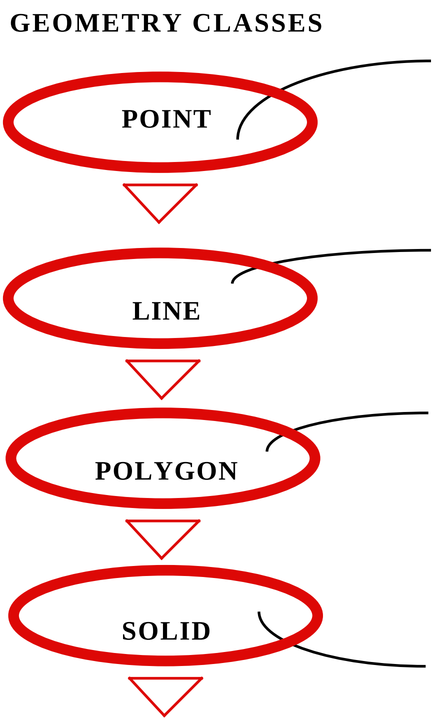
2.1.2 Kit-of-parts Philosophy

The active use of kit-of-parts philosophy was an important element in the conception of VBuild. A kit-of-parts is a collection of discrete building components that are pre-engineered and designed to be assembled in a variety of ways, much similar to an erector set or toy Lego blocks. When assembled, the entire kit-of-parts can define a finished building or artifact. The components fit together according to rigorously designed interfaces which provide for flexible configurations. Components are sized for convenient handling or according to shipping constraints. Since a well designed component can be used over and over again, fabrication processes can be worked out in advance for real-time manufacturing at time of need.

Kit-of-parts components can be thought of as objects in an object-oriented programming environment. With well-defined interfaces which are rigorously followed, the component itself can assume any form. Interfaces can include mounting points, rules for the transfer of loads, specifications for thermal performance, and maximum cost constraints. In short, a kit-of-parts approach lends itself to cheaper and more efficient manufacturing, and is a clean way of demonstrating a network-based virtual building system without having to deal with loose ends associated with the complexities of unorganized raw materials.

2.2 VBUILD CLASSES

The programming classes or objects which have been devised for the browser mostly fall into three main categories: Geometry classes, Assembly classes, and Construct classes. The Geometry classes consist of classes which define geometrical representations of objects, and include 0D, 1D, 2D, and 3D geometry. The Assembly classes define specifications, function, and fabrication processes associated with individual components in a kit-of-parts. An assembly would be a discrete component which is manufactured using a combination of different fabrication processes, and follows interface rules for connection to other components (in this paper, “assembly” and “component” may be used interchangeably and refer to the same thing). Construct classes define ways of organizing the assemblies.

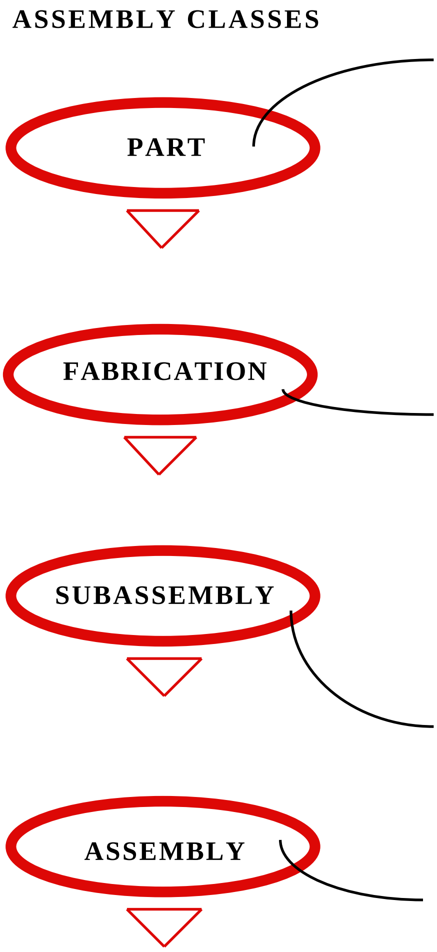


The POINT class represents the most basic form of geometry or 0D geometry. One point object would contain XYZ coordinates as data, and include various methods for manipulating the coordinates.

The LINE class represents 1D geometry and includes both lines and segments. A line will have two point objects as data and include methods for manipulating itself.

The POLYGON class represents 2D geometry of enclosed shapes. Eventually it will also represent compound 2D elements such as splines, curves, and open shapes as well, but for now only closed shapes constructed of a series of connected vertices is implemented.

The SOLID class represents 3D geometry. Eventually it will also represent curved surfaces as well, but for now only faceted solids are implemented.



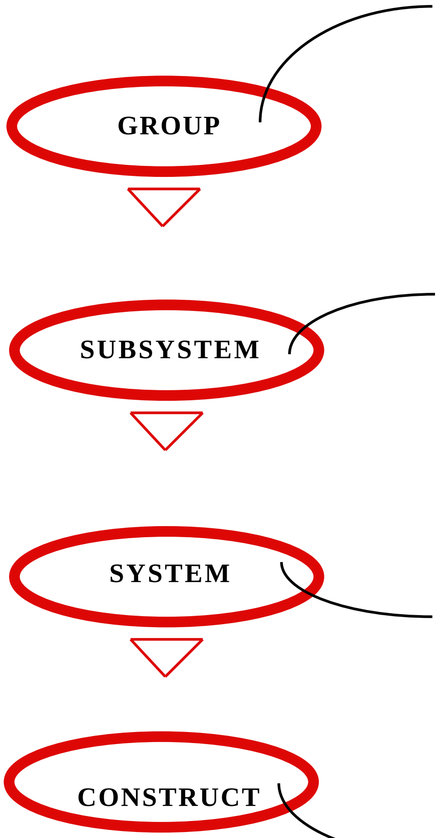
The PART class potentially contains one solid or one polygon or both, as well as a material definition. A part is meant to represent a raw material with limited definition of a potential shape. The solid would be representative of a cast object or folded sheet fabrication, where the polygon would represent a section for extruded objects or the shape of a cutout from sheet stock.

The FABRICATION class has exactly one part as well as paths for extrusion, and includes methods for creating G-code for actually machine fabricating the part (G-code defines paths for manufacturing tooling). Implementation of these methods is currently underway, and will always be added to as new fabrication methods are adopted.

The SUBASSEMBLY class can have many fabrications, as well as links to sensors and actuators that potentially can be embedded in the assembly / component.

The ASSEMBLY class represents a kit-of-parts entity. It can have many subassemblies and would be the largest object that would potentially be manufactured off-site. Each assembly of a certain type only exists once and is instantiated as many times as necessary. The instance would represent real manufactured components assembled on the site.

CONSTRUCT CLASSES



The GROUP class is a special collection of instances of assemblies which have some common purpose or function, such as all the column assemblies of a certain type that belong to a certain floor of a building. A group can have many instances. Methods in the group class would manipulate groups.

The SUBSYSTEM class is a collection of groups which have a common purpose or function, such as all the columns of a building. Each floor may have its own group of columns, which added together would constitute the column subsystem. The subsystem would contain methods which would be for the express purpose of manipulating subsystems.

The SYSTEM class is a collection of subsystems, such as the structural system of a building. The structural system would include column subsystem, floor framing subsystem, and foundation subsystem. System class methods would manipulate, view, or analyze entire systems.

The CONSTRUCT class is the finished artifact, which includes all the systems. Construct methods would define behavior of the construct, and include mechanisms for viewing it in various ways.

In addition to the three main groups of classes, VBuild also has attribute classes and utility classes for the viewing and general manipulation of the various data types. In each step of the hierarchy, methods specific to that data type are implemented. More methods can be added later as the need arises.

2.3 PROPOSED VBUILD CONFIGURATION

VBuild will actually consist of several programs functioning in unison in various locations on the Internet. The browser Plug-in portion is merely the local tool which helps the user view and manipulate the data. Locally the user will create a construct which can be saved as a file locally. Assemblies / components will be returned on request from remote kit-of-parts virtual librarians. Fabrication of real parts will be coordinated by a virtual contractor, and remote control and monitoring will be facilitated by a virtual facility manager. Except for the plug-in, each of the virtual servers will consist of simple Common Gateway Interface (CGI) programs, which are small programs linked to web pages that perform simple automated tasks. The process is delineated in the three steps of design, manufacture, and facility management.

2.3.1 Design

When the user wants to insert another component, the VBuild plug-in will contact remote virtual librarians which will return requested components according to type. The assemblies / components for the most part are high-level parametric primitives consisting of collections of cuboids, cylinders, cones, frustums, and other solids that can be defined with a limited amount of data. Since the amount of data is small, Internet transfer can occur very quickly. The local plug-in then fills in the rest of the data according to a predetermined formula based on the type of primitive and creates an instance of the component. Each time the same type of assembly is requested by the user, another instance is created from the previously downloaded data. When the user saves the construct, the actual data of each assembly / construct is deleted and only the instance references and their locations are preserved. Each time that model is opened, the real data associated with each instance is once again downloaded in real time from the source. Using filters, the user will be able to view the data in various ways which may include sections and plans.

2.3.2 Manufacture

Once the building is virtually assembled, the assemblies / components can be manufactured and delivered. Each assembly / component can consist of many fabrications. Since the fabrications contain the data necessary for their own manufacture, the VBuild plug-in will contact a virtual contractor and request an estimate based on fabrication type. The contractor will select a manufacturer based on price and wait list and return an estimate. The ideal setup would have a simple list on the manufacturer's server which would hold current setup and processing rates, with another list on the material supplier's server which would hold current material costs. A queue could also be maintained on the manufacturer's side which would hint at possible wait lists. When the manufacturer finishes processing the fabrication, it will be shipped directly to the designer or another designated address (such as the site). In the case of multiple fabrication types in a single component, a system of bar codes will facilitate the forwarding of one finished fabrication to the next manufacturer, who will build upon the previous work until the entire component is built and sent to the site.

2.3.3 Facility Management

Once the building is actually assembled, a Hypertext Transfer Protocol (HTTP) or remote access server can be paired with a virtual facility manager CGI on a local computer installed in the finished building. An HTTP server would be used to facilitate an Internet connection, as opposed to a remote access server which supports private phone connections. Each subassembly has the capacity to link to a CGI program which can handle bitwise communication with a LonWorks, CEBus, or X-10 interface device connected to the computer's serial port. LonWorks, CEBus, and X-10 hardware are brand name pseudo-standards which facilitate plug and play monitoring and control of other devices on a powerline network or dedicated bus. These devices can plug into a standard 110 volt outlet, and send and receive signals along the power wire to and from other similar devices. The user would be able to fly-through the VRML model of the building and click on various parts of the model to affect monitoring and control. Upon clicking on the link, special CGI programs would bring up a Java control panel especially prepared for that device.

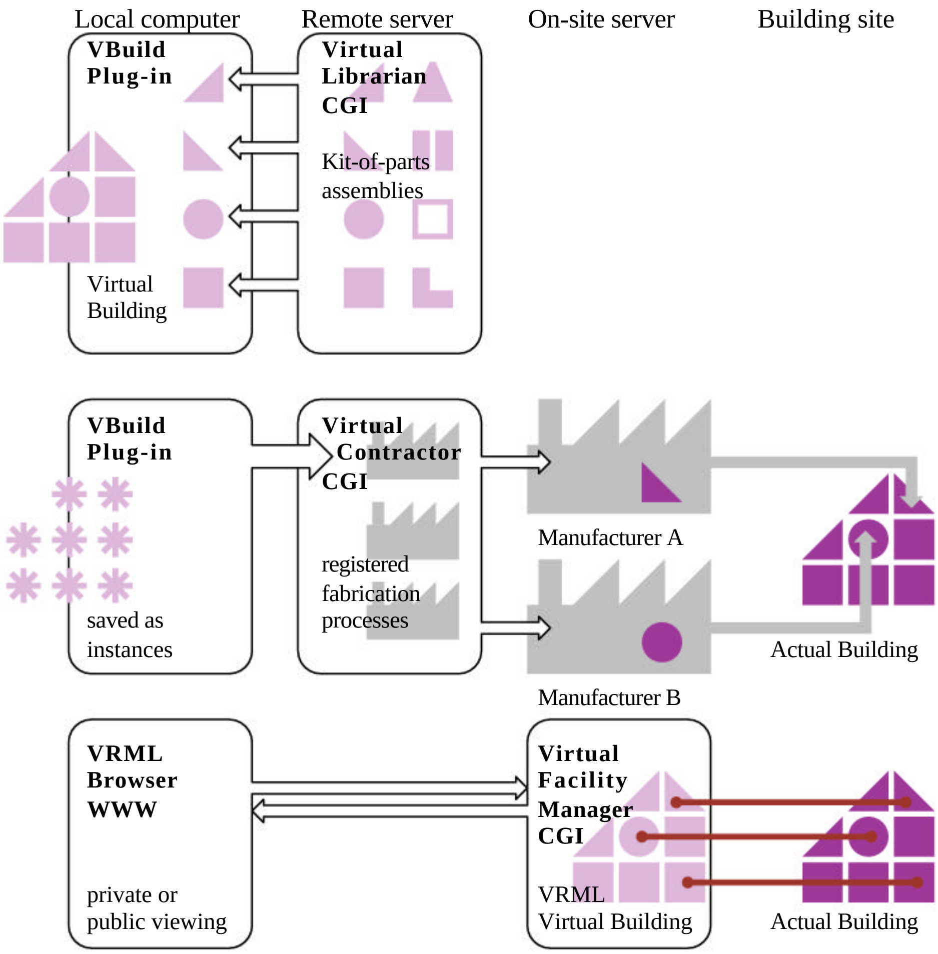


Figure 1: VBuild Proposed Implementation

Figure 1 describes the processes associated with the three phases of design, fabrication, and facility management. In the design stage a designer on a local computer would install the VBuild plug-in in a World Wide Web (WWW) Internet browser such as Netscape. VBuild would allow the designer to collect kit-of-parts assemblies and assemble them together in an intuitive way much the same way a child would construct an object out of Lego blocks. The assembled virtual building could then actually be manufactured component by component through the brokerage of the virtual contractor which would have various manufacturer's fabrication processes on register. Once the actual building is built, a VRML version of the virtual building would be loaded into a computer installed in the actual building, and would have access to CGI's which provide an interface to the LonWorks, CEBus, or X-10 hardware (symbolized in Figure 1 by lines connecting the virtual building and actual building). Full Internet access to the VRML virtual building could be facilitated for public access, or limited phone connection only could be facilitated for remote private use.

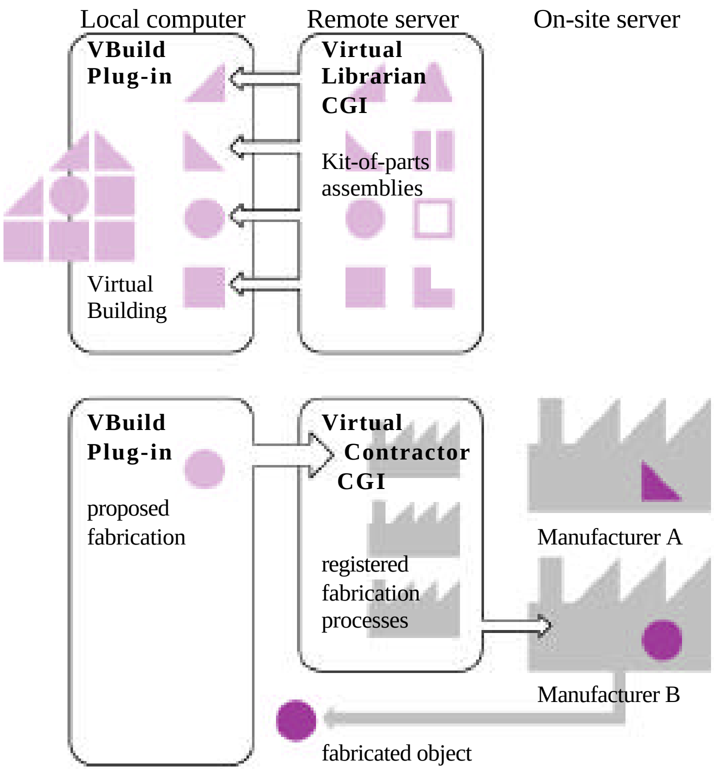


Figure 2: VBuild Current Implementation

2.4 CURRENT VBUILD IMPLEMENTATION

Currently the VBuild plug-in has skeletal appendages of most of the proposed implementation. All geometry classes are complete and operational. The assembly classes and construct classes are present in a skeletal form with limited implementation. Data structures, including linked lists, arrays, and overall file formats are fully implemented, with several experimental parametric primitive assemblies prepared for insertion into a sample kit-of-parts library. Conversion of GEDIT PH3-SET's into full blown VRML models (complete with dummy lighting and camera nodes) is implemented.

The code for actually selecting a component from the web and inserting it into the model has yet to be written, but a demonstration application is being prepared. None of the final CGI programs have been prepared as of yet, but simple implementations have been written and tested. A quick implementation of a link to a manufacturer has been developed and tested.

3. Related Research

The VBuild project is supported on a triple foundation of extensive research in the areas of architectural information visualization, automated construction, and remote facility management.

3.1 INFORMATION VISUALIZATION

During the course of information visualization research, a simple experimental network-based kit-of-parts library was established using the VRML standard. The experiment consisted of three simple virtual building components located on a server in Japan, with a building model located on a computer in Michigan. Using a VRML web browser the Michigan model could be opened, whereupon the components were automatically loaded from Japan in real time and placed in their proper locations and orientations (see Figure 3). Clicking on a component downloaded its specification page. During the course of the visualization research, valuable insight was gained in understanding VRML structures and overall Internet File Transfer Protocol (FTP), Hypertext Markup Language (HTML), and hypertext linking and anchoring concepts. Past experience with design and computer modeling proved to be a valuable resource in the research.

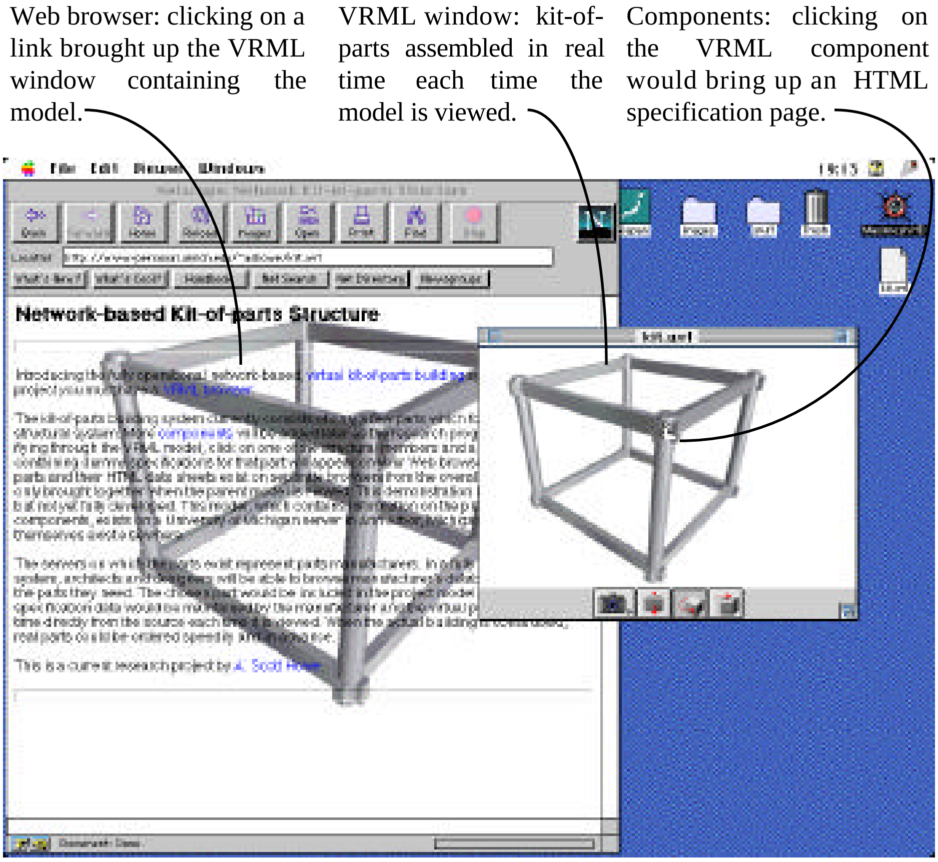


Figure 3: Architectural Information Visualization

3.2 AUTOMATED CONSTRUCTION

An exhaustive study of the state of the art of automated construction was conducted, followed by controlled simulations using industrial robots. A model kit-of-parts building system was devised for the purpose of deriving principles of design for automated construction. The simulations involved the automated assembly and disassembly of the model kit-of-parts from remote locations over the Internet. The components incorporated mechanisms which could interlock with other components that were actuated for deployment by the robot's end affector (see Figure 4). In addition to numerous successful simulations conducted from various computers located on the University of Michigan's local network, control of assembly and disassembly of the Michigan-based model was satisfactorily affected from both Denmark and Japan. Control of the robot was facilitated by the use of a laptop computer connected by modem to the local telephone system using Compuserve telnet. The simulations and construction of the model and components resulted in the derivation of a set of design principles which can be used to design building components with characteristics that easily lend themselves toward automated construction processes. The research also contemplated the development of shape grammars that would not only act as a basis for the design of the building components, but would guide the design of robotic construction machines as well. A simple shape grammar was devised based on the derived design principles and an example conceptual kit-of-parts building system presented, including concepts for robotic systems that could be used to assemble the components on site. The example system illustrated possible applications of the automated construction design principles. Past experience with robotic work cell simulation and conceptual robotic construction system design contributed to the study. The automated construction research provided valuable insight into general principles of design for kit-of-parts systems which are manufactured and constructed using automated means.

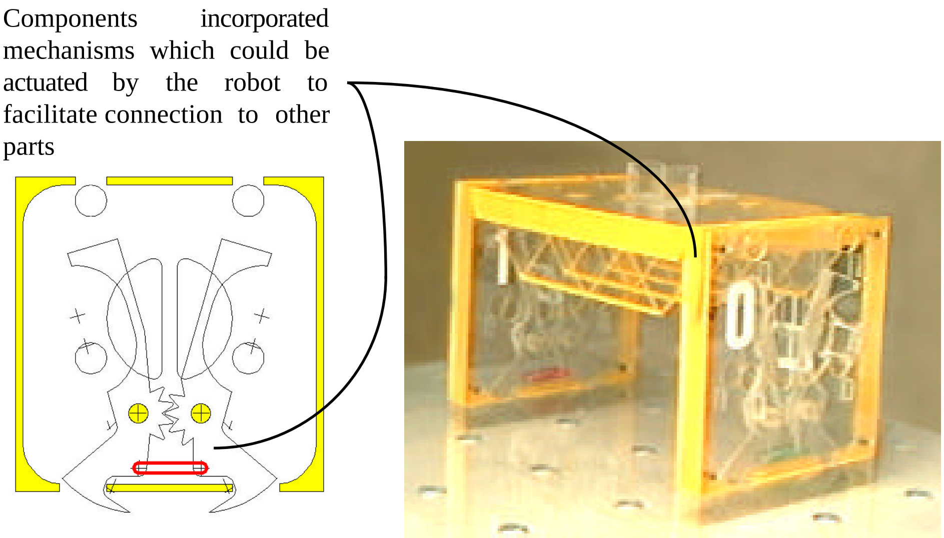


Figure 4: Model kit-of-parts building

3.3 FACILITY MANAGEMENT

The remote facilities management research involved the study of techniques used for remote device monitoring and control. As an experimental project, a facility web page was devised which acted as a home page for an experimental environmental controls facility at the University of Michigan which was wired with various sensors. The web page contained a VRML model of the facility complete with representations of virtual sensors (see Figure 5). Clicking on a virtual sensor would display a recent history of the data downloaded from the actual sensor, using tables and Java graphs. The excercise involved writing a CGI program which linked to the web page. When viewers of the web site initiated a request, the CGI would search through data dumped from the sensors and compile a web page displaying the requested parameters. The remote facilities management research contributed to an understanding of CGI programming structure, Java applets, and a general knowledge of “plug and play” device control and data collection systems such as LonWorks, CEBus, and X-10.

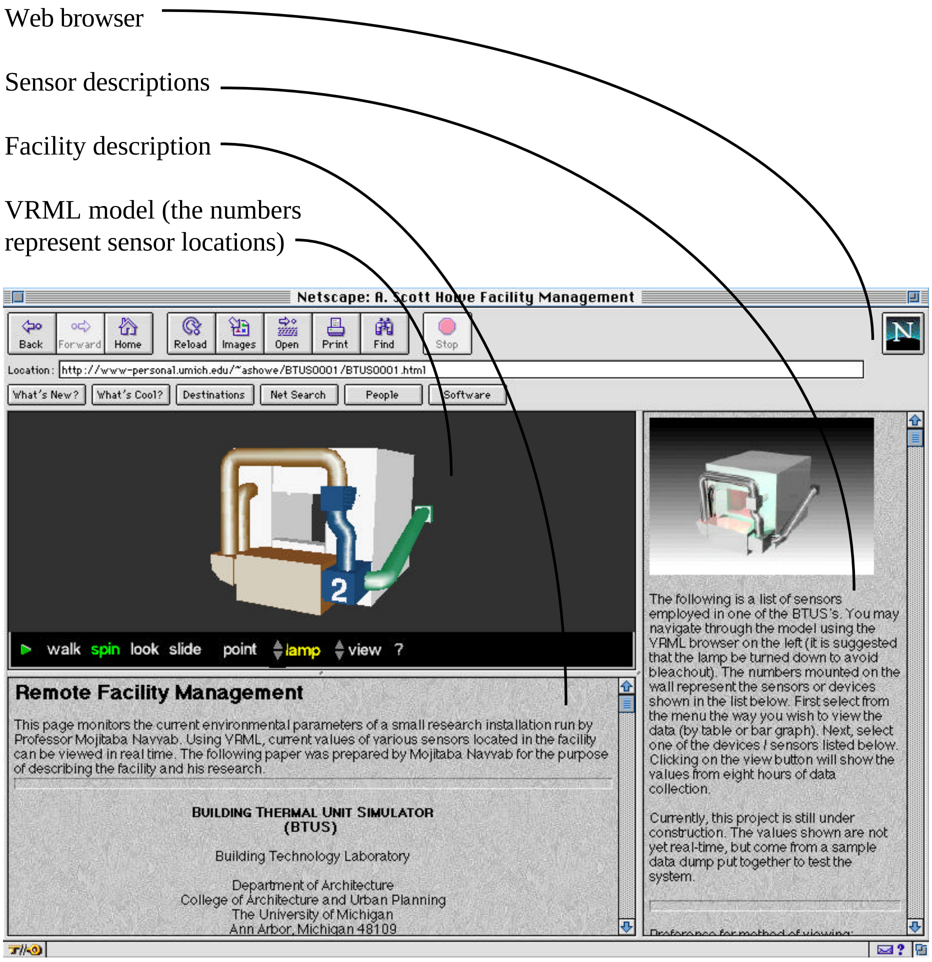


Figure 5: Remote facility management web page

The VBuild project began with a series of C programs which were designed to demonstrate simple network-based modeling functions. A CGI program was written which allowed Internet users to create and view various solid primitives by interactively specifying their parameters on an HTML form. The solids were created by the CGI and returned to the web page for viewing. This excercise proved to be a vital link between the earlier network kit-of-parts experiment completed in the architectural information visualization research and the CGI programming done in conjuction with the remote facility management studies. With the simple web-based solid modeler in place, a study of C++ and the GEDIT modeling classes commenced. The C++ studies included a general familiarization of object-oriented principles and the analysis of advanced data structures such as trees and linked lists. The switchover from C to C++ proved to be extremely valuable in the overall object-oriented / kit-of-parts concept conceived from the beginning. With the C++ foundation, the VBuild classes could be implemented one at a time in order of complexity.

Starting with the POINT class which uses only built-in data types for XYZ coordinates, a series of methods were devised for manipulating points. The methods included creation of point, deletion of point, calculation of distance between points, and the transformation of points. A GEDIT utility class, 3D MATRIX, was adopted early on and facilitated transformation functions. The LINE class built upon the point class, using points as data types representing end points. Methods included the transformation of lines, calculation of segment length, and the equal division of segments. The POLYGON class adopted GEDIT's PG3-SET data structure and borrowed some of its algorithms, such as area and normal calculation. Many other algorithms were added from scratch, such as perimeter calculation and the line extraction function. The SOLID class also adopted GEDIT's PH3-SET data structure and fuctions such as the calculation of volume, surface area, and centroid. Other algorithms were added from scratch, including polygon extraction and spread centroid calculation. In the same way, each of the classes were defined and tested using special programs designed to load data and use each of the methods.

The object-oriented structure of C++ proved to be a tremendously powerful programming foundation. After each class was implemented and tested, the objects they defined could be completely relied upon. In the hierarchical structure of VBuild, the dependence upon lower level classes made rock-solid reliability a necessary requirement. In addition, the opportunity to build upon the tremendous amount of work put into the GEDIT modeling functions by staff and students of the University of Michigan Architecture and Planning Research Laboratory contributed tremedously to the functionality of VBuild.

4. Future Research

Proposed work will initially continue with the implementation of various unfinished functions. A robust interactive web interface will be implemented which will allow the designer to intuitively add, move, position, and delete assembly instances from the model. DXF interpretation will be facilitated, and all CGI programs will be put in place. Once the program is in a usable state, the plug-in will be made available to selected designers.

As a test for the experimental design environment, a small model kit-of-parts library will be prepared with complete fabrication processes incorporated in each component. In addition, an X-10 device will be included in each component's subassemblies. With data prepared, the full sequence of a design / build / monitor simulation scenario will be initiated and documented, using real manufacturers and designers.

Other possible extensions to the experimental software could include collaboration with other researchers and organizations. The development of shape grammars that embody typical configurations and connections could function as generic components much similar to the way ASCII text relates to different font styles. The shape grammar would define the interfaces between components, and the component itself could be designed to a certain style. A family of such components would form a library “font style”. Designers would quickly be able to switch between styles due to the fact that each of the libraries are designed according to the same shape grammar. Another possible avenue for joint research could be the development of web-based analysis software and expert systems for evaluating the performance of virtual buildings and constructs. Expert systems could take information based on the underlying shape grammar and inform the designer of inadequacies or violations of pre-engineered parameters.

5. Conclusion

For simplicity’s sake kit-of-parts philosophy is utilized in this work. When individual designers begin to use the system they may want to be able to define their own components rather than use those already designed by someone else. Along with the ability to harness various fabricatioin processes for the purpose of facilitating the manufacture of a pre-defined kit-of-parts, real time design and manufacture could be a next step. Regardless of whether the design consists of extemporaneously thought-out elements or pre-designed components, the ability to create a virtual artifact and link it back to the real one should for the purpose of design, manufacture, and facility management prove to be a powerful tool.

The eventual goal would be to have a real building that adapts to the users needs through user-initiated instructions affected by the manipulation of its virtual building counterpart. Initial construction and renovative construction could eventually utilize robotic systems which are either brought in from elsewhere or are actually incorporated into the components of the building itself. Redesign and renovation would be facilitated by changing the virtual building to meet the new needs, and executing the automated construction and assembly features to bring about the changes in the actual building. Though this research does not directly address the use of automated construction, it contemplates its eventual use and attempts to forsee possible preparations for future seamless integration.

NOTES & REFERENCES

A. Scott Howe, architect, is a designer for Kajima Corporation of Tokyo, Japan and a PhD candidate at The University of Michigan School of Architecture and Urban Planning. For more information see Home Page URL: http://www-personal.umich.edu/~ashowe

VRML REFERENCES

Mark Pesce (1995) *VRML: Browsing & Building Cyberspace*, New Riders Publishing, Indianapolis, Indiana.
John R. Vacca (1996) *VRML: Bringing Virtual Reality to the Internet*, Academic Press, London.

OBJECT-ORIENTED PROGRAMMING REFERENCES

Gregory Satir, Doug Brown (1995) *C++ The Core Language*, O' Reilly & Associates, Inc., Sebastopol, California.
Mark Allen Weiss (1996) *Algorithms, Data Structures, and Problem Solving with C++*, Addison-Wesley Publishing Company, Menlo Park, California.
John December (1995) *Presenting Java*, Sams.net Publishing, Indianapolis, Indiana.
Suleiman Lalani, Kris Jamsa (1996) *Java Programmer's Library*, Jamsa Press, Las Vegas, Nevada.
Thomas Boutell (1996) *CGI Programming in C & Perl*, Addison-Wesley Publishing Company, Menlo Park, California.
Apple Computer, Inc. (1994) *Inside Macintosh: Devices*, Addison-Wesley Publishing Company, Menlo Park, California.
James Turner, et.al., *GEDIT solid modeler*, The Architecture and Planning Research Laboratory, University of Michigan, Ann Arbor, Michigan.

KIT-OF-PARTS REFERENCES

Robert W. Marks (1960) *The Dymaxion World of Buckminster Fuller*, Reinhold Publishing Corporation, New York, New York, pp88-91.
Stephen C. A. Paraskevopoulos, Harold Borkin, et.al. (1968) *Research on Potential of Advanced technology for Housing*, Architectural Research Laboratory, The University of Michigan, Ann Arbor, Michigan.
Udo Kultermann (1970) *Kenzo Tange: Architecture and Urban design 1946-1969*, Praeger Publishers, New York, NY.
Moshe Safdie (1970) *Beyond Habitat by 20 Years*, Tundra Books, Montreal, Quebec.
Emilio Ambasz (April 1972) *Italy: The New Domestic Landscape, Achievements and Problems of Italian Design*, The Museum of Modern Art, New York, NY.
Kisho Kurokawa (1977) *Metabolism in Architecture*, Westview Press, Inc., Boulder, Colorado.
Frank Russell (1985) *Richard Rogers + Architects*, St. Martin's Press, New York, NY.
Toshio Nakamura (May 1988) "Norman Foster 1964-1987", *A + U Architecture and Urbanism*, Extra Edition.
Colin Davies (1988) *High Tech Architecture*, Thames and Hudson, London, Great Britain, pp42-55, pp68-85.
A. Scott Howe (1989) "Intelligent Living" and "Intelligent Office" in: *Kawasaki Safety Intelligent Plaza*, University of Utah Master of Architecture Thesis, Salt Lake City, Utah: University of Utah.
Wim J. Van Heuvel (1992) *Structuralism in Dutch Architecture*, Uitgeverij Publishers, Rotterdam, The Netherlands.
Martin Pawley (1993) *Future Systems: The Story of Tomorrow*, Phaidon Press Limited, London, England.
Vittorio Magnago Lampugnani (1995) *Renzo Piano: Progetti e architetture 1987-1994*, Electa, Milano, Italy. Book in italian.

LONWORKS, CEBUS & X-10 REFERENCES

V. Elaine Gilmore (September 1990) "U.S., Japan, Europe: The World's Smartest Houses", *Popular Science*, pp56-65, 102.
Denny Radford of Intellon Corp. (November 1996) "Spread-spectrum data leap through ac power wiring", *IEEE Spectrum*, Vol33 No11, pp48-53.

VIRTUAL BUILDING REFERENCES

James Turner, et.al. (August 1990) "AEC Building Systems Model", *ISO TC/184/SC4/WG1, Document 3.2.2.4*, The University of Michigan, Architecture and Planning Research Laboratory, Ann Arbor, Michigan.
James Turner (May 1991) "Guide to Reading NIAM Diagrams", The University of Michigan, Architecture and Planning Research Laboratory, Ann Arbor, Michigan.
A. Scott Howe, Hirata Yasutoshi (July 1991) "A Feasibility Study for a New Architectural Design Approach Using 3D Solid Modeling CAD Systems", *Fourth International Conference on Computing in Civil and Building Engineering*, Tokyo, Japan.
Edward F. Smith (Summer 1992)"Virtual Buildings: Knowledge Based CAD Models for Design, Analysis, Evaluation and Construction", *Computer Solutions*, pp30-32.
A. Scott Howe, Hirata Yasutoshi (1993) "Architecture, Urban Planning Example 1: Design Using 3D CAD", *IBM CATIA: State of the Art 3D CAD / CAM, Technical Papers '92 / '93*, Tokyo, Japan pp6-9, 36-40.

INFORMATION VISUALIZATION REFERENCES

George W. Furnas (April 1986) “Generalized Fisheye Views”, *Human Factors in Computing Systems CHI ‘86 Conference Proceedings*, Boston, pp16-23.
Benjamin B Bederson, Larry Stead, James D. Hollan, “Pad++: Advances in Multiscale Interfaces”, SIGCHI ‘94 short paper.
Sougata Mukherjea, James D. Foley (April 1995) “Visualizing the World Wide Web with the Navigational View Builder”, *Proceedings of the 3rd International World-Wide Web Conference*, Darmstadt, Germany, Computer Networks and ISDN Systems; v 27 n 6, pp1075-1087.
A. Scott Howe (May 1996) “Internet-based Architectural Visualization”, presented at the *ACSA European Conference*, Copenhagen, Denmark.

AUTOMATED CONSTRUCTION REFERENCES

Roozbeh Kangari, and Daniel W. Halpin (1983) “Potential Robotics Utilization in Construction” , NSF research report under grant no. CEE-8319498, National Science Foundation, Washington, D.C.
Leonhard E. Bernhold, Dulcy M. Abraham, and Davis B. Reinhart (April 1990) “FMS Approach to Construction Automation” in *Journal of Aerospace Engineering*, Volume 3, No.2, pp.108-121.
Taisei Corporation (2 November 1990) “Announcing the Age of Construction Automation, Now!” in: *Nihon Keizai Shinbun* newspaper article.
Miroslaw J. Skibniewski, and Stephen C. Wooldridge (1992) “Robotic materials handling for automated building construction technology” in: *Automation in Construction volume 1*, Amsterdam, Elsevier, pp.251-266.
Akinaga Makoto (7 June 1993) “Four Firms Start Construction in the Search for a New Architectural Manufacturing Paradigm” in: *Nikkei Architecture*, pp.160-165. Article in Japanese.
H. Kurita, T. Tezuka, and H. Takada (May 1993) “Robot Oriented Modular Construction System - part II: Design and Logistics” in *Automation and Robotics in Construction X*, proceedings of the 10th International Symposium on Automation and Robotics in Construction (ISARC), Houston, Texas, Amsterdam, Elsevier, pp.309-316.
Colin Bridgewater (1993) “Principles of Design for Automation applied to construction tasks” in: *Automation in Construction volume 2*, Amsterdam, Elsevier, pp.57-64.
Kajima Corporation (1994)“AMURAD Grow-up System” in project report, also conceptual robotic system by A. Scott Howe, Tokyo: Kajima Corporation. Report in Japanese.
A. Scott Howe (1994) “A Genesis System” in: *Special Research Report of the Sensitivity Engineering Product Development Research Group: A Sensitively Designed City*, Tokyo: Japanese Ministry of International Trade and Industry, pp.73-92. Report in Japanese.
Japanese Ministry of Construction (1994) “Development of Automated Systems for the Construction of Reinforced Concrete Structures” in: *Ministry of Construction Technology Development Project Collection: New Construction Technology Development for the Construction Industry*, volume 2, Tokyo, Japan Ministry of Construction, pp.14-95. Report in Japanese.
Javier Ibanez-Guzman (1995) “Modeling of on-site work cells for the simulation of automated and semi-automated construction” in: *Construction Management and Economics*, pp.427-434.
Intelligent Manufacturing System (IMS) Collaborative (1995) “Automated Materials Delivery System for High-rise Buildings” in product pamphlet, Tokyo, IMS Collaborative. Pamphlet in Japanese.
A. Scott Howe (April 1997) “Designing for Automated Construction”, presented at the Second Conference on Computer Aided Architectural Design Research in Asia, Hsinchu, Taiwan.

REMOTE FACILITY MANAGEMENT REFERENCES

Jeffrey M. Hamer (1988) *Facility Management Systems*, Van Nostrand Reinhold Company, New York, pp79-84.
Ken Sakamura and Richard Sprague (April 1989) “The TRON Project”, *Byte*, Vol14 No4, pp292-301.
H. Michael Newman (1994) *Direct Digital Control of Building Systems: Theory and Practice*, John Wiley & Sons, Inc., New York, New York.

SHAPE GRAMMAR REFERENCES

William Mitchell (1990) *The Logic of Architecture: Design, Computation, and Cognition*, MIT Press, Cambridge, Massachusetts.
Shuenn-Ren Liou (1992) “A Computer-based Framework for Analyzing and Deriving the Morphological Structure of Architecture”, doctoral dissertation, The University of Michigan.