Applying web-based product libraries

Richard Coyne*, John Lee, David Duncan, Salih Ofluoglu
Department of Architecture, University of Edinburgh, 20 Chamber St., Edinburgh, Scotland EH1 1JZ, UK

Abstract

This paper presents progress on a research project about on-line libraries of product information as used by architects, engineers and other design professionals. We present product library assistant intranet, PLA(id), which is a system programmed in Java for organising product library information on the World-Wide Web (WWW). PLA(id) is to be used experimentally in practice contexts to elicit insights into the applicability of on-line libraries of product information. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: CAD; On-line libraries; Product libraries; Digital libraries; Multimedia network applications; Java applications

1. Introduction

This paper follows on from a report presented at EuroplA97 on an EPSRC-funded project examining how design practitioners are using, or could use, product information on the World-Wide Web (WWW) [2]. By “product information”, we mean catalogues of information about standard constructional components and materials of the kind found in the Sweet’s Catalogue and the Barbour Index used by architects. Product information also includes two-dimensional and three-dimensional geometrical data of the kind presented in CAD libraries. Both kinds of information are now available on the Web, variously circulated and distributed by CAD suppliers, product manufacturers/suppliers, service providers and CAD users. In the CAD area, Bentley System’s Microstation (http://www.bentley.com/products/elinks/mslink/), Autodesk’s AutoCAD (http://data.autodesk.com/) and Graphisoft’s ArchiCAD (http://www.objectsonline.com/) provide CAD library information (with geometrical and other attribute data) on the WWW [1] and product suppliers such as Algoma (USA — http://www.algomahardwoods.com/) and Andersen (USA — http://www.andersenwindows.com/) (Fig. 1) have extensive on-line catalogues of windows and doors on the WWW (some in PDF format). In the service area, there is First Source (USA — http://www.afsml.com/products/) and the RIBA (UK — http://www.productselector.co.uk/) product catalogues on-line. Some architectural firms also use the Web for exchanging information about products. We are examining the value of each of these modes of distribution of product information to practitioners, determining what tools are needed to make the pro-
cess more effective, and how computing in practice is adapting to the emerging environment of distributed product libraries.

2. The emerging environment of distributed product information

Here, we identify several key issues highlighting what is required for effective Web-based product information to be useful to designers, specifiers and documenters.

2.1. Pointer management

There is the issue of finding, updating, and otherwise managing relevant WWW pointers (URLs) to product information. Current tools include search engines, Web page download tools, and bookmark managers. On PCs and Macintosh computers, pointers to web pages can be turned into icons and organised into directories as files. Some PC and Macintosh spreadsheet, database and word-processing programs recognise URLs (web addresses) and activate a browser to access the page when they are clicked. So it is possible to manage URLs in spreadsheets, databases and tables. In addition, browsers can be customised to efficiently cache web pages and handle subscriptions to web pages. How can such tools be used effectively by practitioners managing product information, and what other tools are needed?

2.2. Distributing pointers

Pointers to web sites can be shared using standard Novell and AppleShare tools, and the Unix directory structure. There are also tools such as Basic Support for Cooperative Work (BSCW, developed at the German National Research Center for Information Technology — http://bscw.gmd.de), which enable users to set up web pages for sharing files and URLs. Participants within design projects expect to be able to share their product information, a concern, which resonates with the emergence of the “virtual office” [5]. How well do current systems of distribution serve the virtual design office?

2.3. Synchronous sharing

Along with the development of computer-mediated communication (CMC) tools generally [7], designers might expect to be able to talk about and manipulate links and view content synchronously, from remote workstations. This involves (i) real-time
discussion over links and their content, and (ii) shared organisation of links.

2.4. Integration into CAD

Most CAD systems make use of libraries of information about standardised components, such as ArchiCAD’s element library system, which uses a programming language, graphic design language (GDL), for describing the geometrical and material attributes of elements such as doors, windows, fittings and furniture. Sizes and other parameters can be adjusted as specified in the library programs, and there is a simple-to-use interface for instantiating components into the current project as part of the documentation and modelling process (Fig. 2). Such systems raise two main issues. (i) There is no standardisation of parameterised descriptions for CAD elements across CAD systems [3]. (ii) To date, few CAD systems seem to incorporate active URLs as attribute data in components. If they did then there would be the prospect of accessing product information not normally included in the CAD library description, such as images, specifications, test data, installation instructions, and other information currently available in on-line catalogues. There is also the prospect of automated updating of geometrical component information across the web.

2.5. Design support and product selection

Product selection is an aspect of design decision making. Designers start to consider products and components before the design is put into the CAD system, and it is an aspect of designing that continues throughout the documentation process. There are
several aspects to product selection as part of the design process.

(i) Designers need to browse product and material choices, and see them in combination. In the context of Web-based product information, there is a need to link URL pointers with the visualisation of products and materials, as with the use of manually constructed sample boards for testing choices and combinations, and focusing discussions with clients.

(ii) As far as product selection is concerned, the final outcome of the design process is a set of drawings/models, specifications and schedules that represent a commitment, or a final instantiation from a range of possibilities. But throughout the design and documentation process, the designer may need to keep options open, make tentative commitments, and prioritise choices. Design support tools for Web-based product information should take account of this need.

(iii) There is a distinction between off-the-shelf products and customised products. Tools for product selection should perhaps take account of the distinction between generic/parameteric elements (such as the category: panelled single door), element descriptions with all parameters instantiated (panelled single doors opening inwards with a width of 75-cm painted green) and individual instantiations of elements that appear in the documentation schedules (door number 26, which may have precisely the same specification as door number 28). Do designers want Web access to the manufacturer’s full range of products, information about specific products, or just information about the instantiation in the current project?

(iv) Product selection is heavily based on precedent. Designers commonly develop “palettes” of favourite products that they use repeatedly on different projects. This case-based aspect to product selection can be reflected in the way product information links are managed, but also in the use of tools for searching previously used products according to key attributes. This brings the product library under the purview of database management and search. It should be possible to interrogate a database of Web-based product information: for example, to find information on all previously used fine wood panelled doors. Such precedents may also be shared amongst the participants of a project, and may also have some market value as a product profile pertaining to particular types of projects. There could be an exchange of portfolios of typical products used in schools, offices, retail and so on.

(v) This opens up the possibility of interpreting geometrical and other attribute data on complex search criteria as an aid to design. This is a much broader issue and beyond the scope of this project.

2.6. Distributed CAD

There is the question of how Web-based product information fits within the projected ecology of distributed CAD-related tools [4]: the full CAD system as comprising distributed, platform-independent Java applets [6].

3. A product library assistant intranet — PL(id)

In keeping with the project objectives, we are currently developing a prototypic web environment that has a CAD element library server with the ability to support and manage multiple distributed clients who require access to heterogeneous multimedia CAD library information on-line. CAD users may use such a system for arranging links to on-line product catalogues. The system, PL(id) — Product Library Assistant (an intranet for designers) — is graphical, flexible and we are testing how it fits the modes of working of a CAD user. We intend that PL(id) be run in a browser concurrently with a monolithic CAD system. The system is written in Java and resolves issues of maintaining directories of information at a central server, passing information between client and server, cross platform compatibilities, and access by concurrent multiple users. The system allows CAD users to call up graphical directories of elements used in their current design project and examine product descriptions and specifications that are available on the WWW. Here, we show how PL(id) addresses the issues raised above.

3.1. Pointer management

PL(id) allows users to organise web links graphically and in folders on a WWW Java canvas (Fig. 3). The icons representing links can be moved around the canvas and organised in folders. The pointers can
also be organised as lists. Users can also select from a palette of icons. The idea is that bookmarks can be organised in ways relevant to a CAD project. In these respects, PLA(id) duplicates the functionality of the bookmark organisation of Netscape or Explorer running on a PC or a Macintosh, but with major differences outlined below.

3.2. Distributing pointers

The link information and icon locations are stored at a server site, and the information can be accessed through any browser by password. This distributed architecture allows groups of people to share relevant links according to project. We are thereby simulating a form of virtual office organisation in which we act as a server organisation, with a fast server (SUN Ultra), with good connectivity to the rest of the Internet and good processing capability, and our practitioner participants access our server through the WWW using their standard 33.6 kbps Internet connections.

3.3. Synchronous sharing

A further advantage of using a server architecture is that multiple users can share the same work canvas. So if more than one user logs on to the same project they can see the current state of the display. They can also see how other users move their cursor and icons about the screen. PLA(id) is therefore amenable to shared discussion and collaboration in conjunction with other CMC tools.

3.4. Integration into CAD

Most CAD systems generate schedules: lists of elements, materials and quantities pertinent to the current project in tabular format. Such schedules also serve as lists of element library instantiations. Schedules of components (doors, windows, cabinets, etc.) can be uploaded to PLA(id). These schedules, which are produced by the designer’s CAD system, are parsed on the server to extract each component’s attributes. This information is added to a relational...
database, which PLA(id) accesses to create directories of icons relating to each component. PLA(id) allows a particular procedure to be followed in order to integrate such instantiations.

(i) The user generates a schedule for the current state of the CAD documentation. So with ArchiCAD, the user would specify, which attributes are to be tabulated, request ArchiCAD to generate the schedule and then save the schedule as a text file or spreadsheet on their local machine (Fig. 4).

(ii) From PLA(id), running in a Web browser, it is possible to invoke a CGI script to upload the schedule to the server.

(iii) The server runs a program that “parses” the schedule data and puts it into a relational database. The method of parsing depends on the conventions of the CAD system. At the moment, the parser is set up for ArchiCAD schedules, but would only require minor modification to handle other formats. PLA(id) therefore produces a searchable database where each record is an instance from the schedule. Typically, a record would contain the name of the element (D1-Tr), its type (DOOR), and other information about size, material, direction of swing and details. Not all this information will be relevant to forming the basis of a Web connection. Just a list of elements would suffice, but the other attributes can be used for search.

(iv) PLA(id) displays an icon in the current project folder for each element (as in Fig. 3), organised in folders according to the type specification. So all doors would be stored in a door folder.

(v) This provides a reasonable basis from which the user can assign Web links. There are three methods of assigning links: (a) typing links in manually to an edit dialogue box (Fig. 5), (b) uploading a bookmark list from the local browser environment (Fig. 6), and (c) searching the database for suitable links that have been used before. See below.

### 3.5. Design support and product selection

(i) Product and material portfolios: PLA(id) also provides the opportunity to use a Java canvas as a display area for arranging samples of building products. It is common practice in architectural and interior design offices to organise samples of materials on a board to test their compatibility, and to show clients. An electronic version of a sample board seems to be a useful and graphically relevant way for designers to organise product information. Clicking on any sample invokes a web site with further information. The user may wish to arrange their own samples from images taken from the web or materials scanned using the office scanner, in which case they can upload images from the user machine onto the server using a CGI file upload utility. Samples can be arranged and sized on the canvas (Fig. 7).

(ii) Keeping options open and (iii) Generic or specific links: This aspect of PLA(id) has not been fully developed yet and requires user studies to develop with any confidence. It is likely that users will prefer to work with collections of links rather than one-to-one links between library elements and Web sites. So the doors icon or folder will contain a collection of links to different product information pertaining to doors for the particular project rather than links between every door in the schedule and

![Fig. 4. A schedule of components as produced by ArchiCAD.](image-url)
Interpreting attribute data: PLA(id) automatically iconises elements from the element library for manipulation and display as in Fig 3. How are Web links assigned? The approach is generally through the use of precedent links. All elements are stored as records in the relational database, with attributes, including URLs, as fields. There is a “prototype” field by which the user can flag that a particular element is to serve as a searchable and copyable instance.

Through PLA(id)’s interface, the designer can type in attributes, as notes, relating to preferred attributes (speed of delivery, cost, etc.) of the potential suppliers of these building components. All of the attributes are then used to retrieve potential candidate product links, through standard query language (SQL) search, from the libraries of links built up through the designer’s use of PLA(id). The designer can access this list to either narrow down or add new links. Once a specific link has been selected for the component, there is an option of flagging the link as a further prototypic case (Fig 8).

3.6. Distributed CAD

PLA(id) functions as a prototypic distributed CAD system in that it makes use of various data structure, manages shared data, has a graphical interface, and takes account of library and project data. It consists of a series of Java applets that run on local computers while maintaining its data on a project-by-project
basis on a server machine. There is a sound basis here for exploring other CAD functionality, including consigning heavy-duty processing and the maintenance of large scale databases to the server machine, and of course drawing and modelling.

4. Projected use scenarios

We are studying three scenarios in which systems like PLA(id) might be used in practice. These relate to different focal groups of users, with different interests, motivations and requirements. Consideration of such scenarios is a fruitful way of exposing possibilities, problems and issues connected with the development and use of on-line product information.

4.1. CAD developer scenario

Here, we envisage that the initiative is taken by CAD systems developers who integrate certain features into their CAD systems to make them network
aware, such as: a scripting language that interprets HTML; a feature to allow access to Web sites as well as local files through the mechanism for accessing library data; some kind of automated searching/browsing/indexing and filtering mechanism to identify relevant data and put it in a usable form for the CAD system; interaction tools — when the user clicks on an object in the CAD work environment a link to relevant data on a Web site is invoked (the link is part of the object’s attribute data); the integration of other CMC facilities accessed through the CAD system interface, such as selecting CAD model objects and gaining synchronous video communication to appropriate personnel; and a facility for automatic download of libraries and software modules as needed from the CAD system supply company.

In this scenario, the CAD system is in contact with the CAD system development site through the Internet. The development site maintains the product information and keeps it up to date. There are many incentives for the CAD developer for this kind of contact: including monitoring of software licensing; automatic download of software upgrades; reporting of bugs and fixes; presentation of promotional material to the user. However, there are questions about the viability of this approach where there is competition between CAD systems developers. CAD developers are already undertaking this kind of development. Autodesk and Bentley Systems, for example, are expanding the Web facilities of AutoCAD and Microstation to include many of the aspects just mentioned. They are able to produce drawings and models in slimmed-down formats that facilitate Internet transmission and integration into other designs at remote sites.

In this context, a system such as PLA(id) can be seen as a way of organising and distributing two-dimensional and three-dimensional library/geometrical data in a CAD format. PLA(id) is an “open” system, in that it is not tied to any proprietary standard. CAD developers are commonly ambivalent toward open standards: if they can get a large sector of the market to adopt their own particular standard, then open systems may not contribute much to their profitability; but if (as is more typical) many users use some other standard, then their own users will be advantaged by being able to combine their system with library components (for example) provided by other developers. We see PLA(id) as contributing to a growing tendency towards more open systems in many areas of computing use. It provides a means of organising information that may come from a variety of sources, but which has to be usable in a given system. We envisage a widening use of open standards supporting distributed CAD, much as one sees the rise in the popularity of XML (extensible markup language) in text processing.

There is a number of options as to how a system like PLA(id) could be used among developers and practitioners. Here, we mention a few examples to indicate the possibilities.

- Practitioners could be encouraged to have distributed systems, such as PLA(id), for organising specific sets of pointers to content provided by developers on web sites. In this version of the scenario,
developers would be acting as publishers, for whom PLA(id) would function as a convenient distribution system.

- CAD providers could run their own PLA(id) server as a practitioner service, allowing flexible usage but with privileged status for their own library data. This involves more management by the provider, but may have the advantage that the provider can easily offer updates to product information, can obtain detailed information about users’ preferences and activities, and can target advertising more effectively.

- PLA(id)-like functionality could be integrated into a CAD system in such a way that the CAD system becomes a sets of Java applets, each providing a particular inter-operating function that can be called on for specific needs. It would then be a simple matter to directly link into PLA(id) to provide an alternative view of the CAD system’s representation of the design — a view that emphasises how design components relate to a wider range of information, alternative possibilities, etc. The PLA(id) prototype already provides such a view, but in this scenario it would become much more explicit.

4.2. Design practitioner scenario

We are currently focusing on this scenario, in which we consider teams of practitioners using computer-mediated techniques as a matter of course in their day-to-day activity, the users of CAD systems who work together on the same projects: architects, engineers, consultants, contractors and manufacturers. Such teams commonly share project data, including CAD models and component libraries. For example: the architect passes floor plan files generated on the CAD system to the structural engineer for documentation of the structure; and the quantity surveyor uses the building plan files for quantification. There are already CAD systems (such as Skidmore Owings and Merrill’s Architectural Engineering System — AES) that aim at full integration, maintaining multifaceted building models that are accessible and useful to consultants and contractors, and include structural analysis capability.

In this scenario, team members develop and share their own Internet or Intranet-based CAD libraries. The team participants may be spatially dispersed, and each has access to their own web server for making CAD and library data available to the other participants. We note that this technology is already commonplace, as is evident in the recent versions of the Apple Macintosh operating system, which include a web server as standard.

Here, the designer is faced with a range of choices. There may be libraries of components and materials particular to the team or the firm (especially if the scenario is based on an Intranet). There is also product information available on the Internet applicable to a vast range of projects. Soon all building product information may be available on the WWW. The problem of searching and filtering such a vast quantity of information is major impediment to realising this potential. It is unlikely that product manufacturers will provide information, or even links to information, about their competitors’ products. So the trend towards service organisations, such as First Source, that provide synopses of ranges of products by different manufacturers will probably continue. It is unlikely that such services will ever be complete or up-to-date, and different projects will have different informational needs. The design team will need tools to help maintain and organise information culled from this huge general resource in ways that (i) relate it to their specific design project while allowing them to share it easily, and (ii) help to structure the overall space of available information so that it will be more easily accessible for future needs.

PLA(id) aims to meet these needs through the functions we have discussed. It can help with (i) the ability of the server database to support direct links between components used in a design (in a CAD system) and further web-based information about those products. These links can subserve the integration within the team, as team members link items in their different areas of responsibility, as well as providing easy access to further information about products. PLA(id) also assists (ii) through its general facility to maintain shared, structured collections of links that filter the amorphous mass of available information into a form that is tailored to the needs and interests of the team, individual or organisation.

Our reflection on this scenario helps to disclose the issue of how product information on the web is located, addressed and organised. Designers may gather URL pointers from advertising, from col-
leagues, the professional press, synoptic web sites (First Source), web searches, and so on. But these sources will only provide access to a fraction of the information on the Web, and more adventitious browsing will probably be needed for finding relevant information: “intelligent” browsing tools for partitioning and navigating the emerging information space.

4.3. Supplier scenario

This scenario involves suppliers maintaining Web sites of product information, which can be accessed by different network aware CAD systems or CAD systems running in concert with Web browsers. One or more service providers (who could be commercial or semi-public) co-ordinate the service and maintain a Web site that contains a directory of links to suppliers. This service provider would be analogous to the one-stop, synoptic product information service such as First Source, but providing information that is more integrated with the CAD system. The service provider mediates between component manufacturers and practitioners, offering the latter particular views of products.

How should the product data be presented, co-ordinated and standardised? Here, there is an opportunity to overcome some of the browsing problems noted in the discussion of the second scenario. The supplier can provide a more usefully structured view of the domain, organised and indexed according to appropriate criteria. However, the service provider is likely to present only a partial view of what is available and the information is unlikely to be specific enough for particular projects (because is not tailored to the needs of a particular designer or team). Tools such as PLA(id) could be useful for coordinating and customising access to supplier’s catalogues. Retrieval mechanisms can be developed that provide selective views of the data maintained (up-to-date) by the suppliers and service providers.

5. User testing

We are about to present these scenarios to practitioners for testing. Focusing on the second (practitioner) scenario, we are planning a series of workshops, after which the system will be modified for placement into selected practitioners’ offices. At this stage, of course, there will be no PLA(id)-using CAD developers or component suppliers with whom the users can co-ordinate. While it may be possible to simulate these experimentally, our initial emphasis will be on an intranet, used by the design team, with information derived from combining pointers to more static external information sources.

Acknowledgements

This work is supported by the EPSRC (GR/L06041).

References