This is the published version:

Pun, Sung Kin, Liu, Chunlu, Treloar, Graham, Langston, Craig and Itoh, Yoshito 2003, Development of a multimedia data acquisition toolkit and its application for exchanging demolished materials, in Proceedings of the Postgraduate Construction Research Conference, RMIT University, Melbourne, Vic

Available from Deakin Research Online:

http://hdl.handle.net/10536/DRO/DU:30005116

Reproduced with the kind permissions of the copyright owner.

Copyright : 2003, RMIT University, School of Property, Construction and Project
Development Of A Multimedia Data Acquisition Toolkit And Its Application For Exchanging Demolished Materials

Sung Kin Pun, School of Architecture and Building, Deakin University, Geelong, Vic 3217
skp@deakin.edu.au

Chunlu Liu, School of Architecture and Building, Deakin University, Geelong, Vic 3217

Graham Treloar, School of Architecture and Building, Deakin University, Geelong, Vic 3217

Craig Langston, School of Architecture and Building, Deakin University, Geelong, Vic 3217

Yoshito Itoh, Department of Civil Engineering, Nagoya University, Furo-cho 1, Chikusa-ku, Nagoya 464-8603, Japan

Abstract
In the development of a web-based information system such as a demolition material management system, a great amount of diversified information on projects should be acquired from particular users located with various computer platforms. This issue is difficult to handle using the limited HTTP form submission, which could lead to inaccuracy of the information and inefficiency of the whole system. This paper describes a web-based graphical user interfaced, dynamic and distributed multimedia data acquisition mechanism, which accepts users’ drawings and retrieval information from the canvas and stores the multimedia data on a server for further usages. Furthermore, techniques and principles needed to construct such a multimedia data acquisition tool are addressed in detail. The application of this distributed multimedia tool in developing a web-based demolition material management system is also described.

1. Introduction
On many occasions graphics and images are used to carry information using web-based communication. As a data medium, apart from its colourful appearance, graphics are easily understandable and can carry information that is hard to be stored in the text form. Some software environments such as the Hypertext Preprocessor (PHP) enable a web browser to use graphical user interfaces to provide interactive programs to the end user. This gives great flexibility to web developers and web users in information distribution. However, it is rather undeveloped in information gathering through graphics compared to the online text data acquisition techniques such as the HyperText Markup Language (HTML) form functions (Powell and Whitworth 2001, Strahl 2002).
At the time being, if a user wants to submit some data to a web server in form of a graphical medium, the graphics must first be uploaded to the web server and then be displayed there. This means the graphics have to be prepared by other software applications, which may be desktop graphics manipulation software. Furthermore, for retrieving the information in the graphics, a graphical recognition application must be invoked by the server to convert the graphics to useful information. Summarily, in order to submit a graphic from a client to the server, three separate steps have to be processed, which are drawing graphics, saving graphics on the server and analysing the image into one system. Therefore, it would be beneficial for all the clients, the server managers, and the system functions if these three steps could be integrated into one step.

Online graphical information submission has diversified applications in practice such as in e-learning, networking meetings, online surveys, distributed information systems, and so on. There are, however, few publications on the online graphical drawing techniques. Recently, research efforts have been given to develop web-based chemistry structure drawing techniques (Advanced Chemistry Department 2003). Graphics play an important role in the construction and building industry. It involves drafting, modelling, visualisation and simulation. Since there is increasingly collaboration among constructions enterprises worldwide, a united working environment including graphics applications is required. Internet-based systems are undoubtedly the best base to carry out all the communication. While one party in the collaboration has made the draft or model online, it can be directly shown or modified by other partners through a web page. Online drafting and modelling are convenient and can also avoid the potential conflicts of different desktop software applications, and software versions in file transmission such as email communication.

This research aims to develop an online graphical data acquisition tool and apply it into the development of a web-based demolition material management system (Pun et al. 2003). This paper is organised as follows. After analysing the data submission methods in the conventional web-based online information systems, the following section presents the needs and advantages of a multimedia data acquisition toolkit. Then, this paper represents a method to using the current distributed computing programming techniques to construct such a toolkit to allow web users to generate and submit web-based online drawing graphics. In addition, potential usages of such a toolkit are demonstrated through the development of a demolition material management system.

2. Online Graphical Data Acquisition Toolkit

2.1 Conventional Online Data Acquisition Methods

Because the web browser is one of the most available software tools despite different types of hardware and operating systems, it becomes a
A popular platform to run internet-based applications (although its original purpose is to display static web page that is composed with HTML). HTML has a range of controls that allow users to have interactions with web servers using text fields, buttons, and checkboxes. Users are able to construct a standalone desktop application appearance web page. With high skill and flexibility of web page positioning, controls are placed in the way that is exactly the same as a desktop application running in a web browser (Yang 2000). Moreover, client side script language such as Java Script and VB Script give more flexibility in appearance and are more dynamic than the single web page. The form-based program is given the power to do some complex computations and basic Graphics User Interface (GUI) actions. However, the form-based program is not a desktop program and thus has some disadvantages comparing with desktop software. This is explained further below.

Firstly, the graphical user interface elements that HTML implements are too limited. Although the most commonly used elements such as buttons and checkboxes are available, applications with more complex requirements are hard to develop with merely HTML elements. Furthermore, HTML does not support a drawing tool like most desktop programming languages.

Secondly, if the web-based application involves the server process, a web page using HTML and script language cannot be used as a 'real' graphical user interfaced application. The web page has to be refreshed after it sends the data to the server. The HyperText Transfer Protocol (HTTP) Post and Get methods are used to support transmitting information between client side and the web server. However, the process does not dynamically update the web page, but sends another request to the web server and receives a new web page to display. There is, although short, a gap between two pages that should be regarded as two frames of desktop software. It is also the reason why HTML cannot support some graphical user interface elements such as the slide bar whose values can be changed continuously and instantly. A web page cannot respond to these elements by refreshing itself frequently according to the change in value of its elements.

Thirdly, there is only text in the message form between a client and the server. A typical HTTP message from client to server consists of the names of the form elements and the values of the elements that are predefined in the HTML source of the web page. More complex information, such as an image, is hard to be represented by merely text messages. It can be handled only through processes such as saving a file. A HTML form supports a file upload element; however, the file must exist. This means that the file must be prepared before the information is sent to the server. Because HTML does not have the privilege to access the local file, the file has to be prepared by the user through specified software. This is impractical for application in the context of drawing and saving images.
2.2 Advantages of a Multimedia Data Acquisition Toolkit

So as to improve the above-mentioned defects of HTML form function, a multimedia acquisition toolkit is proposed and implemented. There are lots of benefits which graphics have over text. For example, a brief engineering drawing to be described within necessary documentation is much clearer than a page of text and numbers. Some complex information such as the structure of an object is hard or even impossible to be described in a text message. Images seem to be the only solution at this moment. Nevertheless, graphics need more memory space to store than text, and thus take a longer time to transfer between a client and a server. However, this is not a problem due to increasingly popular high-speed Internet connections (Jia 2000). Other techniques, including image coding and compression, can also help to reduce the traffic load on the Internet. From a technical point of view, a multimedia data acquisition toolkit allows users to draw graphics inside the web browser, which is the client side of the system. When a user finishes the drawing, the graphics can be saved as an image in the server. The image can be shown to others via a web browser. As a result, the traditional text message based communication is promoted to graphical message based communication. This is suitable for the situation when a user wants to share information that may be hard to be described in a plain text message.

Using state-of-art distributed client runtime environments, such as Java, programs are stored on a server and transmitted to clients when they make a request. The process of calculation and manipulation of the program happen in the client machine. In other words, these web browser-based programs do not need to be installed in advance, and are more convenient than desktop software. Another technical advantage is that these programs dynamically update themselves in a single web page, without refreshing the whole page. Different with the forms in HTML, most graphical user interface elements are provided and can be customized in client programs.

3. Multimedia Data Acquisition Toolkit Development

The techniques in developing a multimedia data acquisition toolbar are related to three areas, which are the web server side, the web client side, and the communication to link these two sides. Figure 1 briefly describes the client-server architecture, and the main components and required techniques in each of these three tiers.
3.1 Server Component

The web server acts as both a client and a server. The PHP is a part of the web server. It gives the privileges of retrieval, modification, insertion and deletion of database items by the system program. It generates HTML codes for the client side according to the results from the database.

PHP code is embedded directly into HTML documents. It is supported by a number of web servers. While a HTTP request is received in a web server machine using PHP, PHP acts as a script. Different with Java Script or VB Script, PHP is running in the server and sufficient privileges are given to access files or databases. PHP written in a web database program generally consists of two parts. The first part is to access a database and retrieve its records. The widespread usage of PHP in the database system mainly comes from its excellent support of a range of database products. The second part of the PHP program is to generate HTML codes according to the records retrieved from a database. These two parts give the web page a really dynamic and interactive information access facility. The web page shown to the user is not static or prepared in advance, but dynamically generated as requested by the user.

3.2 Client Component

A client side program is required to run inside the web browser that is the most available tool to access the Internet. Its application entities include HTML, Java Script and Java Applet. HTML gives the appearance and formation of the web page, while Java Script helps in the formation of the web page and validating the data inputted by the user. These two elements communicate to the server using Get and Post methods from the HTTP protocol, and form the main application parts of database access.

Java is used to construct the client component of the distributed drawing toolkit. Java is a relatively new language but it has been rapidly developed
all the time. Java is naturally distributed and is ideal for developing the Internet-based application. Its other features such as object-oriented, dynamic, robust and secure make the language popularly supported, especially by most web browsers.

The main functionalities of the toolkit are to draw the graphics, save back to the server and analyse the graphics. Among these processes, drawing on a board runs on the client side only. Java provides rich Application Programming Interface (API) to support graphics processing (Moller and Schwartzbach 2003). It includes drawing lines, rectangles, circles and other customized shapes. It is relatively simple to develop a tool that allows users to draw simple lines and shapes on the screen.

Java Script is mainly used to control the HTML form. While it is running on the client side, Java Script has the strong ability to access the form elements in the web page. Java Script is able to access the value of form elements and control the form submission. Moreover, there is a facility that allows Java Script to call the function of a Java Applet. This gives the substantial flexibility to the communication between the server and the client. Finally, Java Script is always used in formation and appearance improvement of the web page.

3.3 Communication between Client and Server

The HTTP protocol supports both Get and Post methods to transmit information back to server from the client side. The Get method is simply placing the names of variables and their values in the end of the URL. It is less secure because all values are shown to the user. Moreover, information cannot be too long due to the limited length of URL. The Post method is sending a message through a byte stream between client and server and it is transparent to the client. In this case, the size of the message could be much bigger. Because the image is drawn on the client side and saved on the server side, the content of the image needs to be sent back to the server. Therefore, a relatively larger amount of message transmission is unavoidable, and the Post method must be chosen for transferring the image. The support of the HTTP protocol in Java is comprehensive. Using an URL Connection class, output streams can be built to a server and byte transfers to a server are enabled. This scenario is exactly the same as the HTTP Post method. A file upload activity using a HTTP form can be simulated exactly. The content of the file is retrieved from the file system, then transferred into a byte stream and finally sent to the server. The server gets the bytes through the stream and reassembles them into a temporary file. This file can then be moved to the desired place using a desired name.

In the case of using JAVA as the client interface instead of the HTTP form, no actual file is read from the local file system. The graphics drawn on the screen are coded and sent to the server directly. The server does the same activity to reassemble received bytes into a file. Different with using the HTTP form submission, the web page shown in the client’s web
browser does not need to be refreshed. However, for completeness of the whole system, the server side script program is constructed so that it gives feedback to the client to indicate the necessary messages such as the completion of the file transmission.

In another point of view, the requirement might need the page to be refreshed as usual in web-based systems. Saving the file to the server can be seen as a single part of the whole HTTP form submission. This can be achieved using Java Script. Even though Java Script is only a subset of Java language, there are mechanisms that allow communication between these two languages. Java Script language can invoke the function in the Java Applet object, and it can also get the returned value from the function. Therefore Java Script can request the saving graphics on the server by invoking specified function defined in Java Applet. Java Script has the fill ability of HTTP form control and saving graphics can therefore be set to act just before the form action.

Java has another advantage to enable saving graphics efficiently. Graphics, represented in computer by image files, have lots of format. The primitive format, which is Bitmap, records colour for every single point inside the image. It takes huge space to store the image and thus much longer time to transfer. Java provides a group of classes that enable JPEG coding of the graphics (Moller and Schwartzbach 2003). JPEG is a well-known and widely used image format because of its high compression of the image. Since the JPEG coding is happened in the client side, less time for transferring the image to the server is needed. Selectively shifting computation load to a client machine is a trend of client-server computing. Efforts have been made in researching about process or program migration such as intelligent agents (Anumba et al. 2002).

3.4 Security Consideration

While every web system is under the potential threat of being broken into, security considerations need to be taken seriously in the development of the toolkit (Kaufman 2002). The potential risk in such a system is that once data are sent out from a client, they can be captured by someone else. If the intruder is able to read and understand the message, the system is said to be weak in security. One simple mechanism of strengthening the security is to encrypt the message. While the intruder cannot understand the message after it is captured, the server and the client can understand the message since they have a secure key. In developing the designed toolkit, the main message transmitted is the content of the graphics, which is same as the content of an image file. To secure the transmission, special coding can be applied to the byte stream. There is a security key related to the coding and the key is held by both client and server. The client uses the key as a parameter to encode the byte stream that represents the graphics. The server uses the key as a parameter to decode the byte stream it receives. The intruder may capture the byte stream. However, without the key, the byte stream becomes senseless. Other security issues such as authentication and integrity can be
addressed in the system that implements the toolkit. The popular mechanisms are web-based authentication and Secure Socket Layer (SSL).

There are several hurdles in implementing online graphics tools in practice. First, graphics take larger storage and thus take longer time to be transmitted from one machine to another. Furthermore, while graphics can be easily understood by human beings, this is not fine for computers. It is hard to index and search the information if there is no text information. There is a need to transfer the main feature of the graphic image into text information and store the information into a database. To overcome these obstacles, more state-of-the-art computing techniques need to be applied. To decrease the size of images created, the client side will code the graphics canvas as a vector image rather than a raster image. After all, lines and shapes are usually used in construction industry, which is ideal to be converted into vector information. To retrieve information from a graphic, graphic recognition techniques are needed. By recognizing the lines, shapes, subtitles and their positions, information can be organized and put into a database for further application.

4. Distributed Drawing System Development In Dmms

4.1 e-Demolition Approach to Constructed Facilities

The demolition of building structures produces enormous amounts of waste materials that in most countries result in significant waste streams. In Australia, the construction industry, particularly in the demolition of constructed facilities, is the top contributor among all industry sectors. In most current demolition projects, a great number of demolished materials are directly sent to landfill after their primary usage due to the difficulties to find their next usage immediately. On the other hand, because of lack of supply of second-hand materials, new and high quality materials are used in construction projects whose design standards can be fitted by the secondary or used materials. Waste-exchange systems are an increasingly widespread solution to this problem (Chen et al. 2003, Neuberg et al. 2002). However, because of the flow nature of the current waste-exchange systems and the demolition procedure, they are inefficient to achieve the goal of waste reduction. The recently created concept of deconstruction rather than destruction for demolishing a constructed facility fails to achieve widespread understanding or acceptance due to various practical limitations (Liu et al. 2003). A web-based Demolition Material Management System (DMMS) is being developed for envisaging the deconstruction implementations in practice and promoting cascading usages of construction materials (Pun et al. 2003).

By traversing through the information flow in current demolition projects and wasted material disposals, it was found that lots of demolition problems arise because there is no communication before the waste is
actually created from the demolition activities. Because the site of demolition activity is very likely to be cleaned up due to the preparation of the new construction project, waste materials can only be kept on the site for a very short period. It is certainly hard for both waste material producers and demanders to “find” each other in a narrow time span. Furthermore, lots of waste materials are unlikely to be reused or recycled to satisfy the potential demander because there are no negotiations before the transaction and thus no requirement to comply. Instead of information exchange happening after the waste materials are produced, information is delivered before the waste is actually produced in DMMS so that the electronic demolition (e-demolition) of constructed facilities is approached before the physical demolition. As a result, negotiations are enabled between the material producer and the material demander. This gives great flexibility and time to both parties. They may therefore change their plans to suit the situation of each other and produce detailed specifications on waste materials. With longer time to prepare and plan, demolition projects, construction projects and transportation can be adjusted to connect to each other tightly.

4.2 Applications of the Distributed Multimedia Data Acquisition Techniques

Same as other waste-exchange systems, DMMS is built on the Internet using the form of interactive web page. Figure 2 shows the interface for a user to input drawings of a building to be demolished as well as other data in forms such as the numbers of windows and doors in the buildings. The drawing toolkit allows a user from a client to draw lines and shapes to describe a project such as the external and internal walls shown in the figure. The dimensional and structural features of this project are then transferred to the database from the drawing in the web server. The drawings can be detailed dependent on the user’s desire. In addition, pictures and videos as well as descriptions of the demolition project can also be submitted using the conventional HTML forms.

This data acquisition technique gives flexibility and convenience to both material producers and demanders. Because most demolition material producers or demanders do not have an engineering background, DMMS is purposely developed to help them estimate the amount and classification of the waste materials before the project is undertaken with the assistance of intelligent applications and other computer aided technologies. In addition, the price of the whole project and each type of material is calculated after the unit price of each type of material is confirmed. The material producers and demanders are only asked to provide simple information regarding to the demolition and construction projects respectively. The waste material producers and demanders then receive the estimated data on the materials as well the cost.
5. Conclusions

In recent years, web-based information systems have been developed rapidly in diversified fields. However, their usage in practice is limited because of the restrictions in data submissions from clients. In this research, a multimedia online data acquisition toolkit has been purposely developed to collect graphical data. Under the support of such a toolkit, a user can provide the web server online drawings from a computer platform through the Internet, so the web server will save the drawings, report the drawing analysis results to the user, and manage the submitted data into a pre-developed database information system. The usage of this web-based data acquisition technique is applied for developing a demolition material management system.

Acknowledgements

The research was supported by a summer scholarship provided by the School of Architecture and Building, Deakin University.
References


Jia, W. 2000, High Speed Networks, Printed by Deakin Print Service, Learning Resources Service, Deakin University, Geelong, Australia.


