

Parametric Web-CAD for Box Culvert Design

Julian H. Kang¹, Byeong-Cheol Lho² and Sang-Rung Choi³

¹Texas A&M University, juliankang@tamu.edu

²Sangji University, bclho@mail.sangji.ac.kr

³Hanseok Engineering Co., Ltd., sangrc@kornet.net

ABSTRACT

This paper presents a development of Web applications to improve the process of a box culvert design and document management using XML and SVG. The box culvert is one of the simple and common structures employed repetitively in road construction. Although parametric computer applications have demonstrated a significant amount of time savings in designing simple and repetitive structures, circulating resultant electronic documents among project participants has not been fully integrated with these applications. We developed a parametric Web application in order to facilitate not only engineering design of the box culvert but also management of resultant documents over the Web. This paper presents how XML Data Island and SVG were utilized to generate engineering drawings and display them in the Web page.

Keywords: Design Automation, Web-CAD, XML, SVG

1. INTRODUCTION

A box culvert is one of the common structures employed in road construction. Over the last decade, many box culverts have been constructed in Korea. The shape of these box culverts is fairly simple and typical no matter where they were built. The design of the box culvert is also quite repetitive and relatively simple. Therefore, the design process of the box culvert can be automated using computer applications. Engineering firms have made an effort to develop a computer application that designs and generates engineering documents and CAD drawings of the box culvert automatically by utilizing design parameters provided by users. These electronic documents, however, were printed on the paper for circulation among project participants until the Internet-based communication became available in the Architecture, Engineering, and Construction (AEC) industry.

The Internet-based communication facilitated to exchange electronic documents asynchronously among people regardless of their physical location. However, e-mail communication seldom helped construction professionals collect information systematically. Frequent revisions of the engineering documents often create confusion among project participants. Engineering firms realized that collecting and managing electronic documents in a central repository would help them organize the lessons learned from the completed

projects. Web-based project management, which stores information in the central repository, therefore has attracted many AEC firms because of its potential ability of building a project legacy database by collecting electronic documents in the central repository. The frequency and cost of the bridge repairs made a public sector of the construction industry, such as the Ministry of Construction and Transportation, also interested in utilizing the project legacy database for improving efficiency in infrastructure management.

However, when it comes to collecting electronic documents in the Web-based central repository, parametric design automation system developed for simple structure in 1990's often demands engineers to transmit the resultant electronic files to the central repository almost manually. These applications were basically designed only to expedite the design process in a stand alone computer and little consideration was given for collecting electronic files and manipulating them in the central repository. Once a structure design is complete and CAD drawings are generated, engineers need to send these electronic files to the server via the Internet-base communication such as an e-mail transmission, which often results in mistakes and confusions. If the entire process from engineering design to document management can be implemented in one system, mistakes and confusions in the process of manipulating electronic documents may be minimized.

Emerging Web technologies such as Active Server Pages (ASP), Extensible Markup Language (XML), and Scalable Vector Graphics (SVG) suggest a new development of a parametric Web application that may design a simple structure, generate vector drawings, and manage engineering documents on the Web browser seamlessly.

2. PARAMETRIC DESIGN AUTOMATION

Over the last decade, various kinds of parametric computer applications were developed and utilized for a simple and repetitive structure design. Hyundai Institute of Construction Technology (<http://www.hdec.co.kr/technology>) developed a steel box bridge design automation system, which challenges an optimum bridge design by integrating the processes of designing the structure, generating engineering reports and AutoCAD drawings. Hangil IT Co., Ltd. (<http://www.aroa.co.kr>) also provides a series of computer applications, such as ARoad, AModel, AShop, and APiece, for facilitating repetitive design of a steel box bridge through generating engineering reports and CAD drawings automatically. ACulvert is developed especially for box culvert design, which generates engineering reports, CAD drawings, and a bill of materials automatically. Construction Consulting Group, Inc. (<http://www.ccginc.co.kr>) provides G-Culvert, which is another parametric computer application for automatic box culvert design.

Although parametric design applications on a stand alone computer have demonstrated a significant amount of time saving in the process of designing simple structures such as box culverts and retaining walls, little consideration was made for manipulating resultant design documents. Engineering firms often depend on a separate document management system for collecting and circulating electronic documents generated by parametric computer applications. In order to avoid disorder in manipulating electronic documents, they sometimes set a protocol to deal with electronic documents. However, no matter how well the protocol is prepared, engineers are apt to make mistakes and create confusion when they handle electronic documents. Printed documents are still frequently utilized whenever they need to share construction information with their customers or government agencies. Omission or duplication of information, therefore, can hardly be prevented, especially when there are frequent revisions in design documents.

Although Web-based project management has demonstrated the possibility of building a legacy database of the constructed infrastructure, the process of transmitting electronic documents generated by parametric design automation systems has not been fully integrated with Web-based project management. In

order to build a legacy database of constructed infrastructure within Web-based project management, some design documents may still have to be transmitted almost manually from the parametric design automation system to the central repository in the Web server. Government agencies therefore seldom believe the efficiency of utilizing Web-based project legacy database for infrastructure management.

3. EMERGING INTERNET TECHNOLOGY

Over the past few years, new Internet Technologies such as Extensible Markup Language (XML), Scalable Vector Graphics (SVG), and Active Server Pages (ASP.NET) have been introduced to the construction industry. These technologies have been designed basically to represent dynamic information on the Web browser more effectively. XML is a text format designed to manipulate large-scale electronic publishing and data exchange over the Internet. SVG is an application of XML, which defines tags for illustrating vector graphics on the graphics. ASP.NET is one of several programming languages used to develop dynamic Web pages that manipulate information retrieved from the database. Combination of these technologies may provide us with a chance to develop a Web-based parametric application that will expedite the design process of simple structures and improve the process of manipulating electronic documents generated in the design process.

3.1 Extensible Markup Language (XML)

XML is a simple and flexible text format derived from the ISO 8879 Standard Generalized Markup Language (SGML). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere. Teague et al [1] defined XML as "a set of formatting rules that define structured information in a software-neutral text file". Since XML defines the structured information in a simple text file, it can be recognized by virtually any software program regardless of computer hardware and operating system. As a result, XML can be used for building a software-neutral project legacy database. The information defined by XML can be transformed into a common Hyper Text Markup Language (HTML) by XML Stylesheet Language Transformations (XSLT). XSLT also transforms XML files to other XML applications such as SVG.

In 1999, International Alliance for Interoperability (IAI) proposed aecXML, which is an XML-based language used to represent information in the Architecture, Engineering and Construction (AEC) industry. IAI expects that aecXML would facilitate information exchange of AEC data on the Internet. Harrod [2] noted

that “the main idea with aecXML is to not only establish some standard ways of structuring building data but also to do it so as to enable automated processing of that data as much as practicable”. Zhua and Issab [3] noted in their study on data integration in construction projects that a well developed XML schema of the classified construction information is one of the critical key issues for successful data exchange.

Tserng and Lin [4] developed an electronic data acquisition model for project scheduling using XML-based information standard. They pointed two major obstacles in gaining efficient access to information of multi-contract projects: 1) the variety of data structures that project participants may use, and 2) the lack of an automatic mechanism for data acquisition. They expected that an automatic data acquisition system might solve these difficulties, and thus achieve effective communication among project participants.

3.2 Scalable Vector Graphics (SVG)

SVG is a language for describing two-dimensional graphics and graphical applications in XML. It allows for three types of graphic objects: vector graphic shapes, images and text. To be scalable, in terms of computer graphics, means to increase or decrease uniformly. SVG graphics are scalable to different display resolutions, so the same SVG graphic can be placed at different sizes on the Web page without sacrificing graphic resolution. The same SVG content can be a stand-alone graphic or can be included inside other SVG graphics, which results in a complex illustration built up by several people. Figure 1 shows a sample SVG code that illustrates 400 pixel wide and 100 pixel high rectangular filled with grey color.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="12cm" height="4cm" viewBox="0 0 1200 400"
xmlns="http://www.w3.org/2000/svg" version="1.1">
<desc>Example rect01 - rectangle with sharp corners</desc>
<rect x="1" y="1" width="1198" height="398"
fill="none" stroke="blue" stroke-width="2"/>
<rect x="400" y="100" width="400" height="200"
fill="grey" stroke="navy" stroke-width="10"/>
</svg>
```

Fig. 1. Sample SVG code that illustrates a simple rectangular

For last few years, active research on utilizing SVG graphics has been conducted in various areas. Baravalle et. al [5] presented an approach to produce pictogram representation of numerical data obtained from scientific computer programs using SVG. They used XSLT to transform XML-based numerical data into SVG representation of the pictogram. González and Dalal [6] presented a web service that allows end-users to specify a database query and visualization of the data as charts and graphs using SQL+D, which is an extension of Structural Query Language (SQL). In particular, they

presented two special display classes of SQL+D displays: graphs and charts, for which the output is a standard SVG file that can be stored or displayed using any SVG-enabled browser.

3.3 ASP.NET

Active Server Pages (ASP) is a technology in which the pages are a mix of HTML and a scripting language, such as VBScript or JavaScript. When a certain ASP page is called, the script is interpreted on the fly and generates additional HTML. This generated HTML is mixed with the static HTML, and the page is sent to the browser. ASP.NET is a new technology that copes with the Microsoft's .NET framework. However, ASP.NET is not a simple upgrade of ASP [7]. Although ASP.NET still employs the request/response model, ASP.NET simplifies the coding model by making it appear as an event-driven programming model. In ASP.NET, script can be no longer mixed with HTML.

4. PROTOTYPE DEVELOPMENT

Our group developed a series of dynamic web applications that configure a box culvert in the Web page using parameters provided by the users, and analyze the box culvert to identify the maximum bending moment and shear forces of its members. These Web applications then calculate the amount of reinforcing steel and generate the engineering drawings. Our group used ASP.NET to develop the dynamic Web pages on top of Microsoft Internet Information Service (IIS). Microsoft Access was adopted to store design parameters. Our group also designed XML Data Island to illustrate engineering reports and drawings. In order to manipulate the XML-based documents along with the associate project information, we decided to store them in the database along with other design parameters. For displaying engineering drawings on the Web browser, SVG is adopted. The transformation of XML representations of engineering drawings into SVG is made through an XSLT.

4.1 XML Data Island

XML is employed to define lines and texts of the engineering drawings because it is best compatible to SVG, which is utilized for illustrating the drawing on the Web browser.

A drawing is composed of several drawing objects, which illustrate floor plans, sections, and details. Our group called a series of XML that defines a drawing object as XML Data Island. Several XML Data Islands are composing one drawing. XML-based engineering drawings are then transformed into SVG codes, which are recognized in the Web browser. Figure 2 shows a sample XML Data Island that defines an engineering drawing. Figure 3 shows XSLT syntax used for

transforming XML-based engineering drawings into SVG codes. A sample SVG code transformed by means of XSLT is shown in Figure 4.

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<drawing>
<line><s>blue</s><f></f><sw>3</sw>
<x1>0</x1><y1>0</y1><x2>650</x2><y2>0</y2></line>
<line><s>blue</s><f></f><sw>3</sw>
<x1>650</x1><y1>0</y1><x2>650</x2><y2>520</y2></line>
<line><s>blue</s><f></f><sw>3</sw>
<x1>650</x1><y1>520</y1><x2>0</x2><y2>520</y2></line>
<line><s>blue</s><f></f><sw>3</sw>
<x1>0</x1><y1>520</y1><x2>0</x2><y2>0</y2></line>
<line><s>red</s><f></f><sw>2</sw>
<x1>70</x1><y1>70</y1><x2>580</x2><y2>70</y2></line>
<line><s>red</s><f></f><sw>2</sw>
<x1>580</x1><y1>70</y1><x2>580</x2><y2>450</y2></line>
<line><s>red</s><f></f><sw>2</sw>
<x1>580</x1><y1>450</y1><x2>70</x2><y2>450</y2></line>
<line><s>red</s><f></f><sw>2</sw>
<x1>70</x1><y1>450</y1><x2>70</x2><y2>70</y2></line>
</drawing>
```

Fig. 2. Sample XML Data Island for engineering drawings

```
<?xml version="1.0" encoding="EUC-KR"?>
<xsl:stylesheet version = '1.0'
xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
<xsl:template match="/">
<svg width="15in" height="15in" xmlns="http://www.w3.org/2000/svg">
<xsl:for-each select="drawing/line">
<g style="stroke: {s}; fill: {f}; stroke-width: {sw}">
<line x1="{x1}" y1="{y1}" x2="{x2}" y2="{y2}" />
</g>
</xsl:for-each>
<xsl:for-each select="drawing/rebar">
<g style="stroke: {s}; fill: {f}; stroke-width: {sw}"
transform="translate(20 20)">
<xsl:choose>
<xsl:when test="no= 3">
<path d="M {x1} {y1}, L {x2} {y2}, L {x3} {y3}, L {x4} {y4}"/>
</xsl:when>
<xsl:when test="no= 4">
<path d="M {x1} {y1}, L {x2} {y2}, L {x3} {y3}, L {x4} {y4},
L {x5} {y5}"/>
</xsl:when>
<xsl:otherwise>
<path d="M {x1} {y1}, L {x2} {y2}, L {x3} {y3}, L {x4} {y4},
L {x5} {y5}, L {x6} {y6}"/>
</xsl:otherwise>
</xsl:choose>
</g>
</xsl:for-each>
<xsl:for-each select="drawing/text">
<text x="{x}" y="{y}" font-family="{f}"
font-size="{fs}"><xsl:value-of select="t"/></text>
</xsl:for-each>
</svg>
</xsl:template>
</xsl:stylesheet>
```

Fig. 3. XSLT file developed for transforming XML-based engineering drawings into SVG-based vector graphics

```
<?xml version="1.0" encoding="utf-8" ?>
<svg width="15in" height="15in" xmlns=http://www.w3.org/2000/svg
xmlns:xlink="http://www.w3.org/1999/xlink">
<g style="stroke: blue; fill: none; stroke-width: 3"
transform="translate(0 0) translate(0 460) scale(1) translate(0 -460)">
<a xlink:href="step6_input.asp?doID=317">
<line x1="0" y1="460" x2="800" y2="460" />
</a>
</g>
<g style="stroke: blue; fill: none; stroke-width: 3"
transform="translate(0 0) translate(0 460) scale(1) translate(0 -460)">
<a xlink:href="step6_input.asp?doID=317">
<line x1="800" y1="460" x2="800" y2="0" />
</a>
</g>
-----
<g style="stroke: green; fill: none; stroke-width: 1"
transform="translate(0 0) translate(400 260) scale(1)
translate(-400 -260)">
<a xlink:href="step6_input.asp?doID=319">
<circle cx="400" cy="260" r="3" />
</a>
</g>
</svg>
```

Fig. 4. SVG-based vector graphics which can be displayed in the Web page

4.2 Database Design

A relational database employed to store and manipulate project information, engineering reports, and XML drawings is composed of several tables shown in Table 1.

Table	Role
tblAccess	Control the user's access privileges to the Web pages
tblBCStandard	Keep engineering specifications for box culvert design
tblBoxCulvert	Keep design parameters for box culvert design, XML data island of engineering reports and drawings.
tblCompany	Keep general information of clients
tblDrawing	Keep general drawing information
tblDrawingObject	Keep XML Data Island of drawing objects
tblProject	Keep general information of projects
tblUser	Keep general information of users

Tab. 1. Database tables.

4.3 Structural Analysis and Member Design

Our group decided to use a commercial structural analysis package, which is very popular in Korea, to analyze the structural stability of the box-culvert to be designed. Parameters for structural analysis are collected via a series of Web pages, and saved in the database. An input file for the structural analysis is then created in the Web server, and the Web application calls the structural analysis package to take the input file and calculate the

maximum bending moment and shear force of each and every member employed in the box culvert. The result of the structural analysis is delivered to the database by the Web application, and is utilized to calculate the amount of reinforcing steel bars to secure structural stability.

4.4 Drawing Generation Module

Web applications that design the reinforcing steel also generate the drawing objects for plans, sections, and details of the box culvert. Various parameters stored in the database are again utilized to generate the drawings objects. The drawing objects are defined by the XML tags designed for this project.

5. SAMPLE BOX CULVERT DESIGN

A sample box culvert of width 5.7 m, height 5.8 m, and wall thickness 0.6 m was designed to test the Web application developed in this project. Our group spent about 20 minutes to complete the design from analyzing the structural stability to generating the drawings, which demonstrated a dramatic time saving in box culvert design compared to the conventional process. Figure 5 shows a Web page for inputting design parameters. Figure 6 shows the Bending Moment Diagram (BMD) and Shear Force Diagram (SFD), which are results of structural analysis. Figure 7 shows the page where the users define parameters for member design. Figure 8 shows one of the drawings generated.

6. CONCLUSION

The parametric Web application introduced in this paper demonstrated to facilitate the repetitive box culvert design and manage resultant engineering documents. As presented with the sample box culvert design, engineers should be able to design simple and repetitive structures over the Web without installing any structural analysis packages or CAD packages in the client computer. Our test indicates that the Web-based design automation system that can be available by hourly charge may help small design firms, which are often in charge of designing simple and repetitive structures, reduce their investment on computer systems and eventually increase their profit. The public sector in the construction industry may also be benefited by the Web-based design automation system in terms of managing constructed infrastructure. Those who need to repair, for example, a box culvert should be able to access to its design information easily because information is stored in the Web-based central repository. New information generated in the process of repairing the box culvert may be appended to the legacy database and utilized for future maintenance.

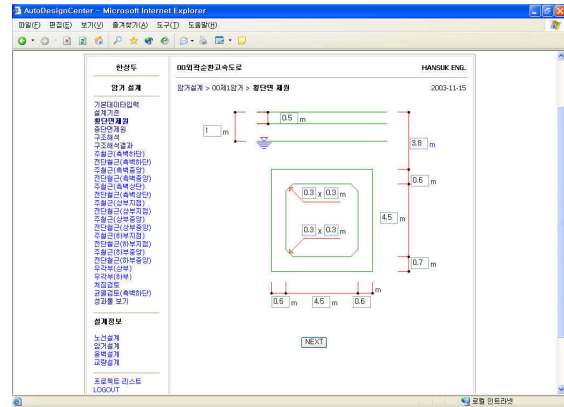


Fig. 5. Web page designed for collecting geometric information of the box culvert

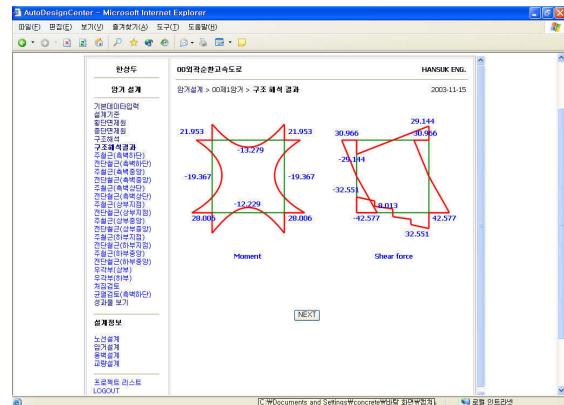


Fig. 6. BMD and SFD displayed in the Web page

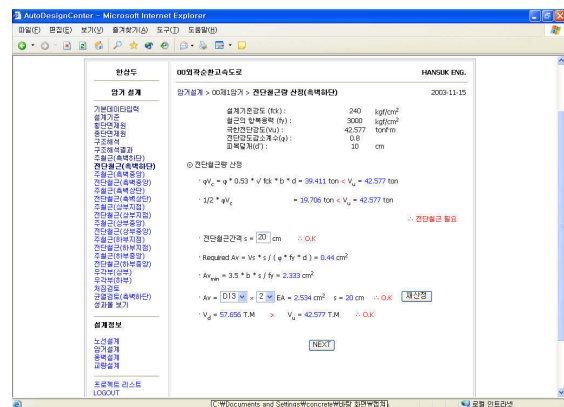


Fig. 7. Web page designed for collecting parameters for member design

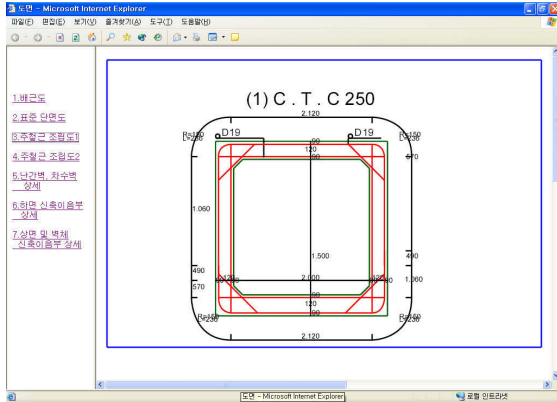


Fig. 8. Engineering drawing generated in SVG automatically

In conclusion, the parametric Web-based design automation system may have a potentiality to shift the process of designing the repetitive simple structures from a stand-alone computing to an Internet-based collaborative computing.

5. ACKNOWLEDGEMENTS

This work was supported by the Ministry of Construction and Transportation of the Korean Government (Grant No. 02 San-Hak-Yeon A01-07).

6. REFERENCES

- [1] Teague T., Palmer M., Jackson R. XML for Capital Facilities. *Leadership and Management in Engineering* 2003; 3(2): 82-85.
- [2] Harrod G. aecXML & IFC. *CADInfo.NET* (www.cadinfo.net); 2003.
- [3] Zhua Y., Issab R. Viewer controllable visualization for construction document processing. *Automation in Construction* 2003; (12): 255– 269.
- [4] Tserng H., Lin W. Developing an electronic acquisition model for project scheduling using XML-based information standard. *Automation in Construction* 2003; 12(1): 67-95.
- [5] Baravalle A. et. al. Using SVG and XSLT for graphic representation. *2nd Annual Conference on Scalable Vector Graphics*; 2003.
- [6] González G., Dalal G. Generating SVG Graphs and Charts from Database Queries. *2nd Annual Conference on Scalable Vector Graphics*; 2003.
- [7] Simmons C., Rofail A. *The Microsoft .NET Platform and Technologies*. Prentice-Hall, Inc. 2002.