

# 벡터 그래픽을 이용한 콘크리트 슬래브교의 설계자동화

Design Automation System for Concrete Slab Bridge using Vector Graphic

노 병 철\*

Lho, Byeong Cheol

김 정 훈\*\*

Kim, Jeong Hoon

김 종 우\*\*\*

Kim, Jong Woo

## Abstract

A Web-based design automation system for concrete slab bridge has been developed to manage the resultant documents as well as to speed up the repetitive design process. Manipulation of engineering drawings in the Web page is one of the critical functions needed for Web-based design automation. Active Server Pages (ASP) is used to that collects design parameters in the database, and eXtensible Markup Language (XML) and XML-based vector graphics are expected to facilitate the representation of engineering drawings in the Web page. In this paper, we present how we used XML and Scalable Vector Graphics (SVG) to compose engineering drawings and represent them in the Web page. XML Data Island we designed to define drawing components turned out effective in manipulating the engineering drawings in the Web page.

## 요 지

본 논문은 콘크리트 슬래브교량의 반복적인 설계과정을 신속히 처리함과 동시에 모든 성과물을 작성함에 있어 웹을 기반으로 한 설계자동화 시스템 개발을 위한 것이다. 웹 기반 구조도면 처리기법은 웹기반 설계자동화시스템 핵심 기술 중 하나이다. Active Server pages(ASP)는 설계 요소를 유용하게 활용하며 이를 데이터베이스에 저장하게 된다. eXtensible Markup Language(XML)과 XML기반 벡터그래픽은 구조 도면을 웹 화면으로 효과적으로 신속히 구현할 수 있다. 본 논문에서는 XML과 Scalable Vector Graphics(SVG)를 활용하여 웹 화면에 구조도면 및 설계 성과품을 작성하였다. 구조도면 작성 시 핵심 도면작성 요소기술로서 개발된 XML Data Island는 웹 기반 구조도면 작성 시 매우 효과적임을 증명하였다.

**Keywords :** ASP, Automation design, Concrete slab bridge, SVG, Vector graphic

**핵심 용어 :** ASP, 설계 자동화, 콘크리트 슬래브교, SVG, 벡터그래픽

\* 정희원, 상지대학교 건설시스템공학부 부교수

\*\* 정희원, (주)에이티맥스 대표이사

\*\*\* 정희원, (주)유디코 대표이사

E-mail : kimjh@atmacs.co.kr 031-706-4328

• 본 논문에 대한 토의를 2008년 4월 30일까지 학회로 보내 주시면 2008년 7월호에 토론결과를 게재하겠습니다.

---

## 1. Introduction

Two-span concrete slab bridge is one of the structures repetitively employed for local road constructions because of its simplicity in design and construction. Every year, a significant amount of simple concrete slab bridges are designed and constructed in Korea. In most cases, the configuration of two-span concrete slab bridges is about the same regardless of their size and topographical features, which in turn makes the design process relatively simple compared to other structures and somewhat repetitive. Design of such simple and repetitive structures as two-span concrete slab bridge is often considered as easy but time-consuming task and therefore assigned to entry-level engineers, which sometimes ends up sacrificing productivity in the design process and reliability of structural stability.

Professionals have attempted to expedite the repetitive design process and secure structural stability by utilizing computer applications that automate repetitive calculations and produce relative documents, which has demonstrated time savings in the course of designing simple structures. However, most of design automation applications have been developed to use in a stand alone computer. The practice of manipulating electronic documents containing the result of calculations has remained the same. Once the electronic documents generated by these applications are tailored and printed for authority's approval, they were most likely managed by those who designed the structure and sometimes disappear in few years as responsible individual relocates. This is one of many reasons why we sometimes can not locate design documents of a certain

structure that needs to be repaired and maintained.

The Internet technology has a potential to better manage those electronic documents produced by design automation applications for a longer duration. If electronic documents are collected in a central repository and retrieved over the Internet whenever needed, document omission and redundant duplication could be dramatically reduced. We may even be able to organize the lessons learned from the previous projects and utilize them to enhance the next project. Web-based project management, which manipulates information in the central repository, has already been utilized to some extent for many Architecture, Engineering, and Construction (AEC) firms to build a project legacy database consisting of engineering reports and CAD drawings and hand it over to succeeding projects. The authorities supervising the construction industry, such as the Ministry of Construction and Transportation in Korea, may also be interested in utilizing the project legacy data to better make decisions for repairing and maintaining the built structures.

However, current design automation applications are not incorporated with Web-based project management. They are basically designed only to expedite the repetitive design process and little consideration was given to manipulating resultant documents. Those who design two-span concrete slab bridges using design automation applications, for example, still need to take additional steps to get the design documents transported to a central repository. Unless a certain system that enforces engineers to send documents produces to the central repository immediately, they tend to put it off until

---

some other day when they don't get pressed by heavy workload. Some engineers may eventually forget to their engineering documents sent to the central repository and move on to a next project. It is reasonable therefore to speculate that sustainable design document management in the cyberspace would not be easily accomplished unless the entire process from the engineering design to the document management is incorporated in one system.

## 2. Design Automation

A variety of design automation systems have been developed and utilized for facilitating the simple and repetitive design.

These applications usually take dimensions of a structure to be designed, loading conditions, and soil conditions of the site and perform a routine computation to check the structural stability with a given structural configuration and generate engineering drawings.

Professionals in Korea have been very active in developing those design automation applications. Hangil IT (<http://www.road.co.kr>) is probably one of the most active software vendors in Korea that have been developing and providing a series of computer applications for facilitating the design of such repetitive structures as box culvers, retaining walls, and simple bridges. It provides, for examples, ARcBridge that is developed to facilitate concrete bridge design. It helps engineers design various types of concrete bridges including portal type Rahmen bridge and slab bridge.

Although these computer applications have demonstrated a significant amount of time savings in the process of designing simple and

repetitive structures, they seldom support resultant document management. Engineering firms often depend on a separate document management system for building project legacy data and use them for succeeding projects. They even prepare a protocol to better collect and circulate the resultant engineering documents. However, the omission and duplication of information still seems to be hardly avoidable.

## 3. Emerging Internet Technology

Recently, new Internet Technologies such as Extensible Markup Language(XML) and Scaleable Vector Graphics(SVG) have been introduced to the construction industry. 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 Web page. Combination of these technologies provides a foundation for developing a Web-based parametric design support system that may facilitate not only the design process of simple structures but also the manipulation of the resultant engineering documents.

### 3.1 Extensible Markup Language(XML)

XML is a simple and flexible text format derived from the ISO 8879 Standard Generalized Markup Language(SGML). It is called extensible because it is not a fixed format like Hyper Text Markup Language(HTML). In 1997, the World Wide Web Consortium(W3C) has announced Extensible Markup Language(XML) to make it easy to use SGML on the Web(2004). Originally designed to meet the

---

challenges of large-scale electronic publishing, XML is also playing an important role in the exchange of a wide variety of data on the Web. Since XML defines the structured information in a software-neutral text file, it can be recognized by virtually any computer systems. Thus, XML can be used for building project legacy data that don't depend on a specific computer system, which is important because no one knows what kind of computer system would be dominantly used in 30 years from now on.

Freedom in defining whatever we like to describe using XML, however, could generate too many diverse descriptions of the same object, which would make it difficult to share one's information with others. Tserng and Lin (2003) pointed out, after they built an electronic data acquisition model for project scheduling using XML, that the variety of data structures project participants used had hinder them from gaining efficient access to information of multi-contract projects. Zhua and Issab (2003) indicated that a well developed XML schema of the classified construction information is one of the critical key issues for successful data exchange. Professionals in the construction industry quickly realized the necessity of establishing a standard to describe engineering documents in XML. In 1999, International Alliance for Interoperability (IAI) proposed aecXML, which is an XML-based language designed for exchanging information in the AEC industry. Harrod (2003) 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".

On the other hand, Autodesk, Hewlett - Packard, Macromedia, Microsoft, and Visio submitted a proposal for Vector Markup Language (VML) to the World Wide Web Consortium (W3C) in 1998 in hopes of presenting CAD drawings over the Web. VML is an application of XML, which defines tags to encode vector information. (Mathews et al. 1998). VML may not be an ideal vector graphics solution for all possible uses, but it possesses attributes enough to display 2D drawings on the Web browser without utilizing specialized plug-in applications. In addition, 2D drawings created by VML can embed hypertext links.

### 3.2 Scalable Vector Graphics(SVG)

SVG is an application of XML, designed for describing two-dimensional graphics in the Web page. SVG graphics are scalable, so the same SVG graphic object can be placed at different sizes on the Web page without sacrificing graphic resolution.

For last few years, various attempts have been made to best utilize the new way of representing vector graphics.

Baravalle et al. (2003) demonstrated the use of SVG for producing a pictogram representation of numerical data obtained from scientific computer programs.

Gonzalez and Dalal (2003) presented a web service that allows end-users to specify a database query and visualize the extracted data as charts or graphs using SVG. Tautenhahn (2002) introduced a 3D SVG library, which enables the Web page developers to display 3D vector graphics using SVG. The 3D SVG library is essentially a collection of JavaScript codes that create a

perspective view of the 3D model using 2D SVG.

## 4. Development of Web-based Design Automation System

### 4.1 System Configuration

Our group has been developing a prototype Web application to facilitate the design automation of box culverts and retaining walls using ASP.NET and XML-based vector graphics technology. Design parameters are collected via a series of Web pages and saved in the database located in the Web server. An input text file for the structure analysis is created from these design parameters. The server application then calls the commercial package and initiates the analysis of the structure. After the structure analysis is finished, the server application reads the output text file of the commercial package to extract the maximum bending moment and maximum shear force.

The bending moment diagram (BMD) and shear force diagram (SFD) are then displayed graphically on the Web page using VML. Once the user provides additional design parameters, the server application selects the amount and type of reinforcing steel bars needed to ensure structural stability. The server application then generates engineering drawings and save them in the database using XML tags. Engineering drawings are converted into SVG and displayed on the Web page. Drawings displayed on the Web page can be printed on the plotter. Figure 1 illustrates this process.

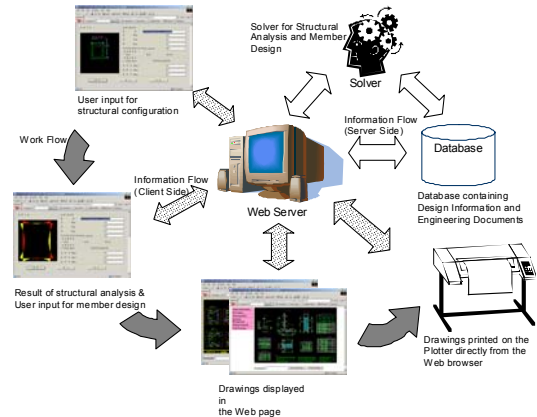


Fig. 1 Process of Web-based design automation

### 4.2 Drawing Composition

An engineering drawing is composed of several components such as floor plans, sections, and detail views. One obstacle, in the process of generating the drawing automatically, is that some of these components may not be depicted properly in the drawing. A floor plan, for example, may be constructed too small or located at the awkward place in the drawing. Modification of the drawing generated by the server application may be an essential function to ensure the quality of the drawing. We decided to add a drawing modification function in a way that users can change the scale or location of the drawing component in the full drawing over the Internet. For this goal, we defined the full drawing as an object that is composed of multiple drawing components. The drawing component is also an object that is created independently with its own origin point. Similarly, the structure object can be defined as a composition of full drawing objects. The project object is composed of multiple structure objects. Figure 2 illustrates

how we composed drawing objects. The local coordinate of the drawing component is converted into a global coordinate when the full drawing is composed. The location and size of the drawing object in the full drawing is first determined by the server application automatically and then adjusted by users. Users adjust the location of the drawing object by assigning an X and Y coordinate in the full drawing, to which the origin of the drawing component will move. The scale of the drawing object in the full drawing can also be adjusted by the scale factor that users will identify. The diagram in Figure 3 illustrates how the full drawing is composed.

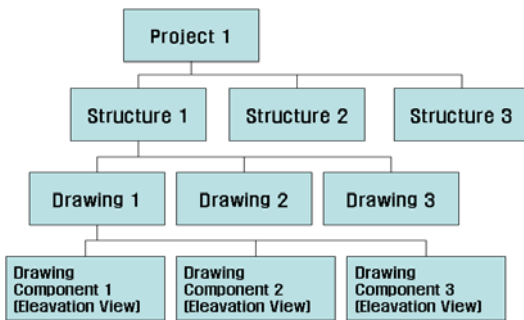


Fig. 2 Structure of drawing objects

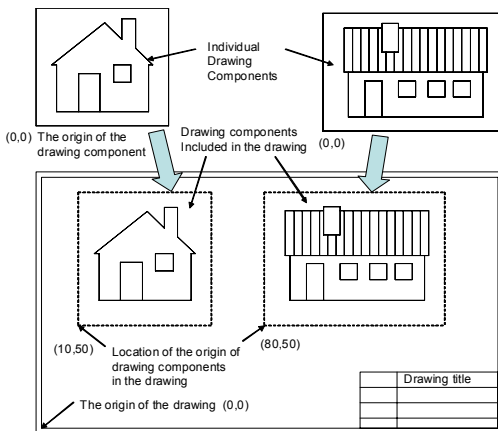


Fig. 3 Composition of drawing

### 4.3 Database Design

Normally, the XML data is stored in a software neutral text file. However, we decided to store the XML Data Island of the drawing component in the database. This decision was made because the XML data for the full drawing was composed of multiple XML Data Islands of drawing components, which could be dynamically changed based on how the user wanted to locate in the full drawing. Accordingly, we designed tables in the database in such a way that the location information and scale factor of the drawing components were stored separately and utilized in the process of composing the XML data of the full drawing. We expect, by saving the drawing components in the database, that we should be able to allow the users to modify the full drawing after it is composed by the sever application automatically. Table 1 shows some of the fields we designed in the database.

The tblDrawing Object table contains the XML Data Island of the drawing component. This table stores 1) drawing component identification number, 2) drawing identification number to which this drawing component belongs, 3) the X and Y coordinates in the full drawing where this component will be located, and 4) the scale factor that will be used to determine the size of drawing component depicted in the full drawing. The tblDrawing table contains general information of drawings. It stores 1) drawing identification number, 2) structure identification number to which this drawing belongs, and 3) other general information of the full drawings. The tblStructure table contains design information of the structure to be designed.

Table 1 Database structure

Table	Information to be managed
tblProject	<ul style="list-style-type: none"> <li>- Project ID</li> <li>- General information of the project</li> </ul>
tblStructure	<ul style="list-style-type: none"> <li>- Structure ID</li> <li>- Project ID that the structure belongs to</li> <li>- Structure type</li> <li>- Design parameters</li> <li>- General information of the structure</li> </ul>
tblDrawing	<ul style="list-style-type: none"> <li>- Full drawing ID</li> <li>- Structure ID that the full drawing belongs to</li> <li>- General information of the full drawing</li> </ul>
tblDrawing Object	<ul style="list-style-type: none"> <li>- Drawing component ID</li> <li>- Full drawing ID that the drawing component belongs to</li> <li>- XML Data Island of the drawing component</li> <li>- Location of the drawing component in the full drawing</li> <li>- Scale factor of the drawing component</li> </ul>

It also stores 1) structure identification number, and 2) project identification number to which this structure belongs.

The tblProject table contains general project information. It may contain the project identification number and the name of project. Figure 4 shows the relationship between these tables. These relationships between tables make it possible to manipulate the drawing component as an object that belongs to the full drawing. In the process of composing the full drawing, the server application extracts the XML Data Island of drawing components from the database and positions them at a proper location in the drawing based upon the size and role of the drawing components.

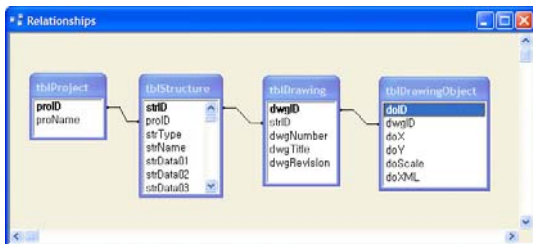


Fig. 4 Relationships between tables in the database

The server application then marks their location in the database and generates the XML data for the full drawing. As a specific component in the full drawing is selected, an editable Web page where the user can update the scale and location information of the component is delivered. If the user provides new location information or scale factor to the database in the server via the Web page as shown in Figure 5, the server application updates the full drawing accordingly. Figure 6 shows how a certain drawing object is updated. The front view, for example, is too small comparing to the adjacent plan view and isometric view when the full drawing was created automatically.

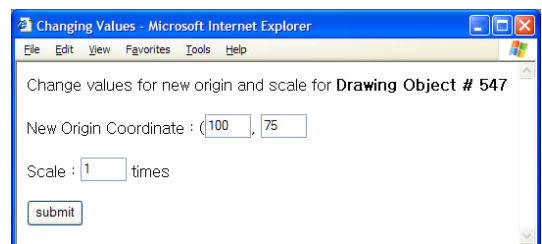


Fig. 5 Web page for updating the drawing component

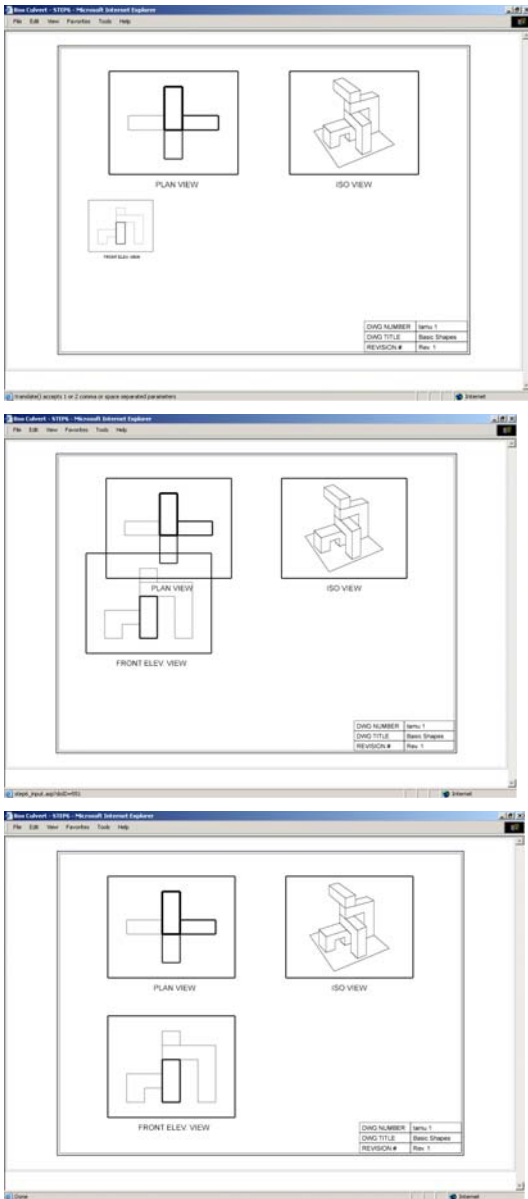


Fig. 6 Update process of the full drawing

Although the user increases the size of the front view, it is not still located at the proper position yet. The user then adjusts the location of the front view to finish composing the full drawing.

## 5. Prototype Development

Our group decided to develop a Web application that would automate the process of designing two-span concrete slab bridges using XML and SVG in order to figure out the potential and limit of these technologies in manipulating electronic documents especially over the Internet.

We first reviewed the current process of designing two-span concrete slab bridges and listed the functions that should be provided in the Web application. Functions that would be needed are: 1) the collection of design parameters, 2) analysis of the structural stability, 3) selection of the optimum structural members, 4) generation of the engineering reports and CAD drawings, 5) representation of the vector drawings on the Web page, 6) generation and representation of the bill of materials, and 7) construction and management of project legacy data. Design parameters collected via the Web page as shown in Figure 7 are saved in the database.

These parameters are retrieved by a server application and tailored into a specific format for structural analysis. For analyzing the structural stability, we decided to plug a commercial package into our Web application.

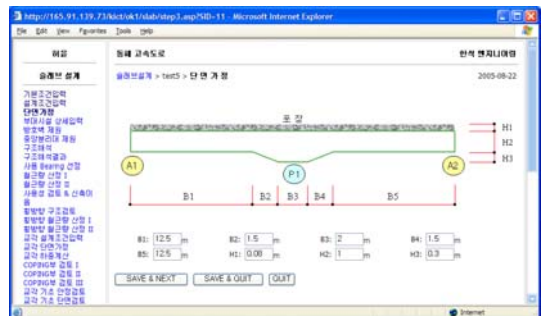


Fig. 7 Web page for collecting design parameters



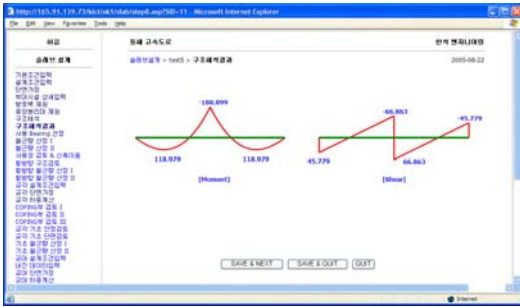


Fig. 8 BMD and SFD generated by Web application

This decision was made in order to take advantage of the reliability of the commercial structure analysis package and to avoid a redundant development. After the structure analysis is finished, the sever application reads the resultant file and extracts the maximum bending moment and maximum shear force. The bending moment diagram (BMD) and shear fore diagram (SFD) are generated in SVG and displayed on the Web page as shown in Figure 8.

After the user provides additional design parameters in succeeding pages, the sever application selects the amount of reinforcing steel bars to secure structural stability. The resultant design information is then saved in the database for the succeeding processes.

```

<line>
<s>blue</s>
<sw>3</sw>
<x1>123</x1><y1>15</y1>
<x2>650</x2><y2>0</y2>
</line>

{Repeat as many as the number of lines
that compose the drawing component...}

<line>
<s>blue</s>
<sw>3</sw>
<x1>650</x1><y1>0</y1>
<x2>240</x2><y2>150</y2>
</line>

```

Fig. 9 Definition of a line in the XML data island

A drawing is composed of several components such as floor plans, sections, and details. In order to allow the users to modify the scale and location of these components in the drawing, we designed an XML data island that defined the lines and texts of each drawing component as shown in Figure 9.

The drawing component defined in the XML data island is stored in the database with associated project information. The scale and relative location information of the drawing component is stored separately in the database. The full drawing is created with a collection of the XML data islands by a server application as shown in Figure 10.

```

<?xml version='1.0' encoding =
'ISO-8859-1'?>
<drawing>

{The collection of XML data island to
be included...}

</drawing>

```

Fig. 10 XML structure of the full drawing

The full drawing illustrated in XML is then transformed into a SVG document according to the XSLT configuration as shown in Figure 11.

```

<?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>
</svg>

```

Fig. 11 Sample SVG tag

Web application we developed worked successfully in the pilot test. A graduate was recruited for the pilot test and he was able to design a 12 meter long two-span concrete slab bridge within 30 minutes using our application. Figure 12 shows one of the drawings generated and displayed on the Web browser.

Although we did not ask that student to design a similar structure without using our application, we speculated that it would take at least entire day for him to produce the same amount of progress.

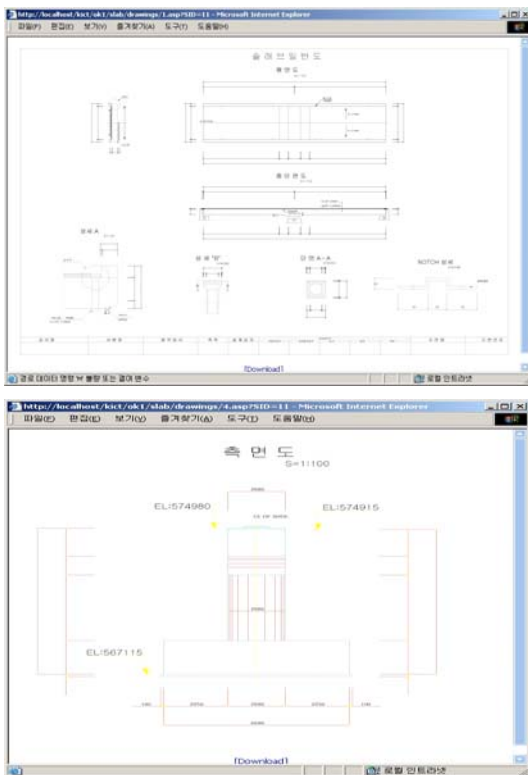


Fig. 12 Drawings generated by Web application

## 6. Conclusions

1) This paper demonstrates how XML and

XML-based vector graphics can be utilized for representing or updating the drawings in the Web-based design automation system. The use of XML Data Island for defining the drawing component facilitates the composition and modification of the full drawing. Although it is not a usual approach to save the XML data in a relational database, the XML Data Island stored in the relational database works effectively in creating the full drawing and updating it over the Internet. Especially, the associated location information and scale factor stored separately in the database facilitates the process of updating the full drawing. Everytime the user updates the location or scale of the drawing component, the server application composes a new full drawing by combining the XML Data Islands and their location information.

2) Manipulation of the drawing component in the sample full drawing demonstrates the potential of a Web-based design automation system in design automation and infrastructure management. Web-based design automation systems are expected to contribute significantly to the creation of the project legacy database in which resultant engineering documents are stored and used for the next project or future maintenance. The drawings stored in the central repository should be a great asset in cultivating collaboration among project participants. The use of Web-based project management has already demonstrated the usefulness of manipulating the project information in the central repository. Accordingly, the Web-based design automation system,

---

which automates the repetitive process of designing the simple structure and save resultant engineering documents in the central repository, should speed up the design process and help the public sector in the construction industry sustain the built-in infrastructure. The only application required to implement the design process is a Web browser. The public sector, such as the department of transportation, is also expected to improve the manipulation of engineering information for infrastructure management by easy access to the right data.

- 3) We believe that the Web application presented in this paper would not only facilitate the repetitive design process of simple structures but also has potential to enhance the process of manipulating the resultant engineering documents. As demonstrated in the pilot test, the entire process of analyzing structural stability, calculating the amount of reinforcing steel, and generating engineering drawings took less than 30 minutes. The time required to complete the entire design may vary according to individual superiority, but we believe that it would take way less than what should be needed with the manual process. The real benefit we are looking for is not coming from saving time in the design process. It would be rather coming from the process of collecting resultant documents in a central repository and retrieve them whenever they are needed. Our application fully automated the process of collecting the engineering document in the central repository. Engineering drawings are generated in XML and stored in the

central repository. They are retrieved from the storage by user's request, converted into SVG, and displayed on the Web browser. As long as the web server maintains, all engineering documents as well as associate design parameters are kept in the server and retrieved by user's request.

We believe that the Web application presented also has potential to help small engineering companies design simple structures without investing in expensive structural analysis packages. If those Web applications are maintained by local authorities and provided to the public on subscription basis, small engineering companies should be able to accomplish their design task fast and cost effectively and local authorities should be able to collect all resultant documents automatically in the central repository and utilize them later on for infrastructure management.

In conclusion, we believe that automating the structural design process and manipulating its resultant documents on the Web environment has a huge potential to shift the process of designing simple and repetitive structures into a whole new paradigm.

#### References

1. Baravalle, A., Gribaudo, M., Lanfranchi, V. and Sandri, T., "Using SVG and XSLT for graphic representation", Proc., 2nd Annual Conference on Scalable Vector Graphics, Vancouver, Canada, 2003.
2. González, G. and Dalal, G., "Generating SVG Graphs and Charts from Database Queries", Proc., 2nd Annual Conference on Scalable Vector Graphics, Vancouver, Canada, 2003.
3. Harrod, G., aecXML & IFC, CADInfo.NET ([www.cadinfo.net](http://www.cadinfo.net)), 2003.

- 
4. Kang, J. H., Park, J., and Lho, B., "Experimental Development of Web-based Building Information Model using 3D VML Library." Proc., International Conference on Construction in the 21st Century (CITC-II), Hong Kong, 2003.
  5. Mathews, B., Lee, D., Dister, B., Bowler, J., Cooperstein, H., Jindal, A., Nguyen, T., Wu, P., and Sandal, T., Vector Markup Language (VML), World Wide Web Consortium Note NOTE-VML-19980513, 1997.
  6. Tautenhahn L., SVG-VML-3D 1.0, <http://home.t-online.de/home/lutz.tautenhahn/svgvm13d>, 2002.
  7. Tserng, H. and Lin, W., "Developing an electronic acquisition model for project scheduling using XML-based information standard", Automation in Construction, Elsevier B.V., Vol. 12, No. 1, 2003, pp. 67-95.
  8. W3C, About SVG. Web Document <http://www.w3.org/Graphics/SVG/About> 4/17/2006 visited, 2004.
  9. Zhua, Y. and Issab, R., "Viewer controllable visualization for construction document processing", Automation in Construction, Elsevier B.V., Vol. 12, No. 3, 2003, pp. 255-269.
  10. 노병철, 김정훈, 이진욱 "2경간 연속 RC슬래브교의 WEB 기반 설계 자동화 시스템 개발", 대한토목학회 논문집, 제27권, 5A호, 2007년 9월, pp. 661-669.

(접수일자 : 2007년 11월 16일)