

Rochester Institute of Technology

RIT Scholar Works

Theses

2006

XPoint: XML, CSS, and HTML-based presentation graphics system

Sudhanshu Sood

Follow this and additional works at: <https://scholarworks.rit.edu/theses>

Recommended Citation

Sood, Sudhanshu, "XPoint: XML, CSS, and HTML-based presentation graphics system" (2006). Thesis. Rochester Institute of Technology. Accessed from

This Master's Project is brought to you for free and open access by RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

XPOINT: XML, CSS and HTML based presentation graphics system

Master's Project Proposal

By

Sudhanshu Sood

Department of Computer Science

Rochester Institute of Technology

Rochester, NY

Chairman: Dr. Axel T. Schreiner

Reader: Dr. Hans-Peter Bischof

Observer: Dr. Fereydoun Kazemian

Table of Contents

SUMMARY	1
1. INTRODUCTION	1
2. XPOINT DESCRIPTION	2
3. ARCHITECTURE	3
4. TOOLS AND TECHNOLOGY	4
5. PRINCIPAL DELIVERABLES	5
6. SCHEDULE	6
7. REFERENCES	7
8. APPENDIX	8
I. APPENDIX A	8
II. APPENDIX B	11
III. APPENDIX C	13
IV. APPENDIX D	14
V. APPENDIX E	15

Summary

Graphical Power Point presentations, when converted to Hyper Text Markup Language (HTML) have a defined graphical format, and are not portable to other browsers. Xpoint is an ongoing development work by Dr. Schreiner. It is a simple presentation graphical system [1]. The presentation is specified in Extensible Markup Language (XML) and is converted to well navigable HTML for browsing, appearance is controlled using Cascading Style Sheets (CSS). The presentation can also be converted to Portable Document Format (PDF) format which provides more flexibility. The project will extend the styles and capabilities of the existing Xpoint graphical presentation system. XML presentation will be presented in various layouts depending on the style template applied. A comparison study will also be done for the portability of the resulting HTML presentations in different web browsers.

1. Introduction

Xpoint is a simple presentation graphics system. The presentation is specified in XML and converted to HTML for browsing. Appearance of the presentation is controlled by CSS [1]. The XML data for presentation will also be converted to PDF (Portable Document Format) for printing/viewing the presentation data. Xpoint is an ongoing development work of Dr. Schreiner [2]. It can be one of the best ways to generate well navigable, portable HTML for simple presentation graphics.

Currently, people use Microsoft power point to generate HTML presentation. Conversion of graphical power point presentation to HTML format does not produce desirable output. The generated HTML presentations by power point are limited to the initial design layout; users do not have a choice to choose any other template during the conversion process. The generated HTML presentation lacks well navigable HTML format. A lot of unwanted HTML code is generated during the conversion process. Moreover the resulting HTML presentation may not be portable to other browsers. The presentation slides can not be converted to PDF format, which is generally desirable as brief snap shots for slides.

The project will focus on generation of portable and well navigable HTML slides from the XML source of a graphical power point presentation. Multiple outputs will be generated from the same XML source depending on the chosen CSS. A PDF document will be generated from the same source for the presentation. A comparison study of the HTML output will be done to study portability issues in various browsers. Since, the presentation data is in XML format we can present data in virtually every possible way.

The project will be sectioned in three areas:

- a. Presenting data in XML format,
- b. Processing and,
- c. Displaying the output in desired layouts and format.

2. Xpoint Description

XML is designed to describe data and to focus what data is. XML allows to create own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations [6]. The data for Xpoint presentation is stored in XML format. The presentation section will have two ways to enter data depending on windows and non-windows platform.

For windows platform, the presentations will be made using Authentic form (Appendix C) and is saved as Xpoint XML format (Appendix A). A form will be designed using stylesheet which will refer to the Xpoint format. This form will validate XML schema and will allow saving an XML file as specified in the schema. XML schema for Xpoint project is available in Appendix B.

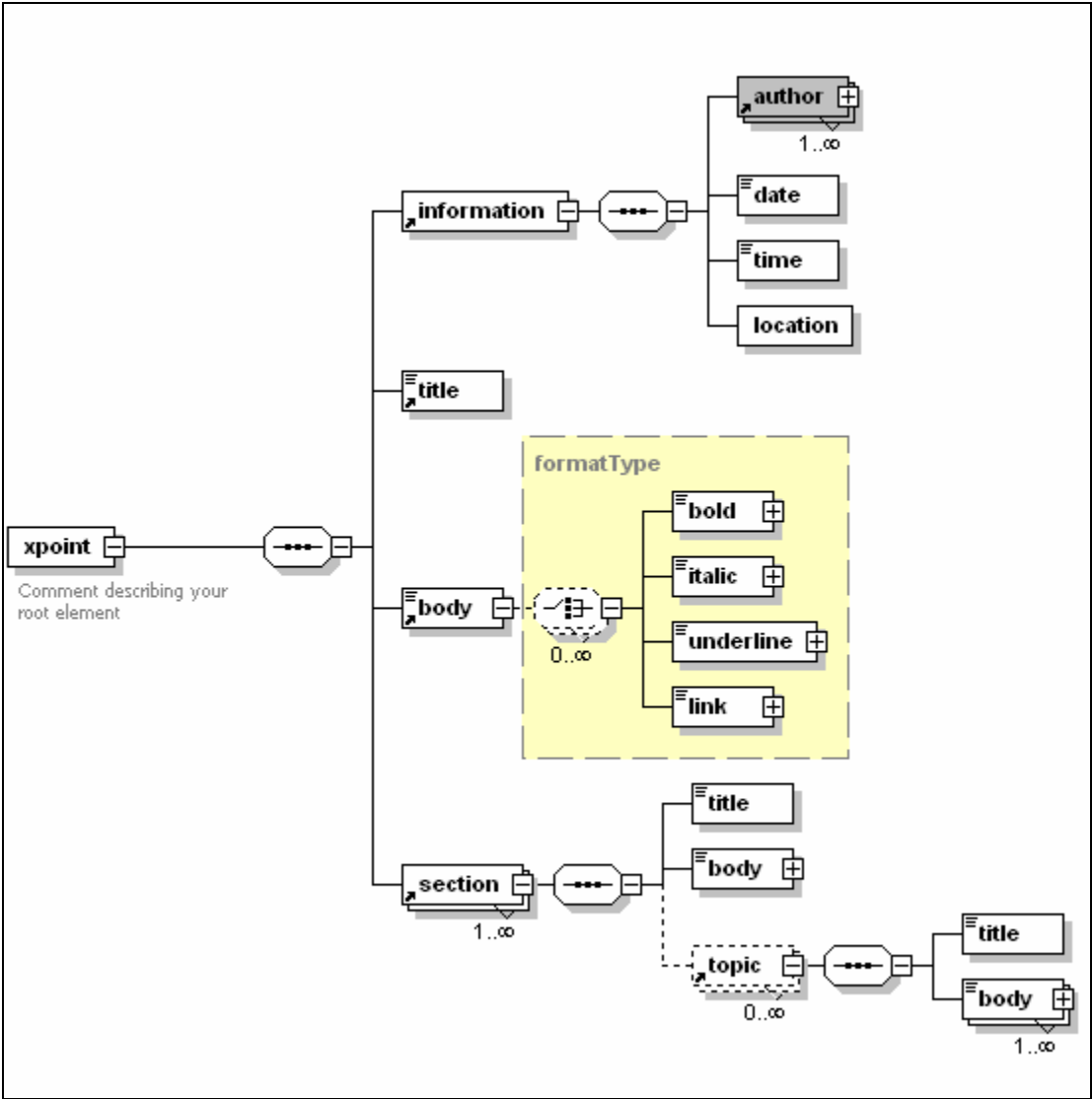


Fig1: Diagrammatic representation of Xpoint XML schema

For non-windows platforms, JEdit will be used to define the presentation and will be saved as Xpoint XML format. JEdit will also validate XML schema and will allow saving an XML file as specified in the schema. JEdit is a cross platform programmer's text editor written in Java [14].

Xpoint contains information, the title page, the body and the sections. Each section contains a title page and some topics. Each topic has a title and one or more bodies. The body can contain arbitrary XHTML markup. As defined in Fig1, the body can have bold, italic, underline and link. An XML Schema defines the elements, child elements and the attributes that can appear in a document. It can also define the order, the number of child elements, the data types for elements and attributes [13]. The complete description of the schema will be specified in the technical report for the project.

3. Architecture

For windows platform, Altova Authentic [11] will be used as a WYSIWYG (What You See Is What You Get) editor. A form will be designed using Altova Style sheet which will validate XML Schema to obtain valid Xpoint XML format. This form can be viewed in authentic editor and is the place where the user can enter information about the presentation. The user will be able to choose for multiple section(s) or multiple topic(s) with in a section.

For non-windows platform, JEdit [14] will be used as an editor. JEdit will be customized by using various Plugins (XML, JTidy) and will validate Xpoint XML schema. Ideally JEdit can be used in any platform (windows or non-windows) but the user will have to compromise with the WYSIWYG feature available in Authentic.

The Xpoint XML obtained from the authentic form will be processed to obtain Navigable HTML and PDF document. Cocoon [9] will be used to generate navigable HTML and PDF document. Cocoon is an open source project under Apache Cocoon project [9]. It is an XML publishing framework. It offers a flexible environment based on pipeline concepts. The configuration file ("sitemap.xmap") for the project allows dynamic setup for pipelines consisting of a generator, multiple transformers and a serializer. Appendix D explains the configuration file, presently being used in the project. It contains information regarding matches, generators, transformers, serializers, list of processing pipelines etc. A Serializer converts an input XML structure into some other format (not necessary XML). Various kinds of serializers are HTML serializer, FOP serializer etc. Various style sheets will be applied to source Xpoint XML format and navigable HTML will be generated. Extensible Stylesheet Language Transformation (XSLT) will be used to generate XHTML/HTML from the source. Various CSS formats will be created to show the flexibility of the output in this project.

Extensible Stylesheet Language – Formatting Objects (XSL-FO) will be used to obtain PDF (Portable Document Format). XSL-FO (Formatting Objects) is a language for formatting XML data for output to screen, paper or other format. Processing will be done in cocoon [7] environment. Appendix E displays a sample XSL-FO file to obtain a PDF

document. The complete design and explanation will be discussed in the technical report for the project.

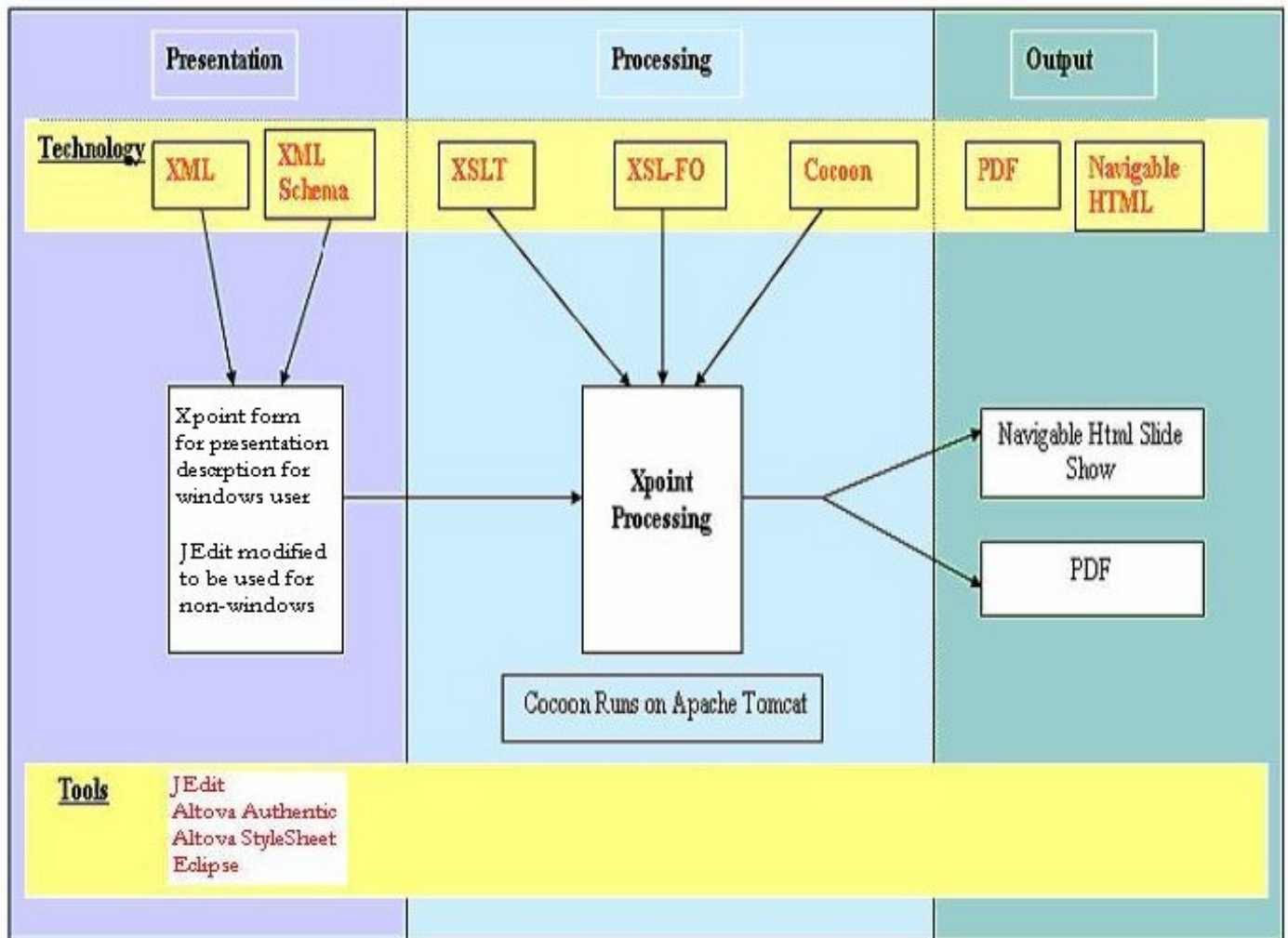


Fig2: Diagrammatic representation for Xpoint Architecture

The project will also compare the portability of the HTML in various browsers. These HTML presentations can be used in an effective way to demonstrate slides. Presentation can be obtained in multiple patterns based on the selected CSS or chosen template.

4. Tools and Technology

1. **Altova Authentic [11]:** Proposed as a WYSIWYG editor to obtain XML format of the presentation specified. Authentic is a free content editor that allows modifying contents in XML document and databases.
2. **JEdit [14]:** Proposed as an XML editor to be used in non-windows environment. JEdit is Open Source software that can be used as an XML editor.

3. **Altova Style Vision [11]**: This is used to design a form for XML input by the user. Form generally agrees to a DTD or XML schema. In this project Xpoint presentation form will be designed using style vision. The form will agree with XML schema. This form can be visualized using Authentic.
4. **Eclipse [10]**: Eclipse is an open source community whose projects are focused on providing an extensible development platform and application frameworks for building software. Proposed to be used for project development, XML, XSLT and CSS.
5. **Cocoon [9]**: Proposed to be used for processing, development and generation of HTML and PDF files. Inclusion of the cocoon.war file to the webapps directory will make the cocoon functional in Tomcat server. XSLT and XSL-FO processing are thus made into a simple process. A map file can be created to specify the transformation of the XML document into the required format.
6. **XSLT (Extensible Stylesheet Language Transformation) [4]**: A language for transforming XML documents into XHTML/HTML documents or to other XML documents.
7. **XSL-FO (Extensible Stylesheet Language – Formatting Objects [12]**: An XML-based markup language describing the formatting of XML data for output to screen, paper or other media.

5. Principal Deliverables

The following are the principal deliverables of this project.

1. A technical report for the design architecture and specifics of this project
2. Examples of navigable HTML presentations and PDF documents for different presentations and comparison in different browsers to study portability.
3. Source code and the files needed to run Xpoint.

6. Schedule

The following indicated are the proposed milestone dates. They are intended to guide the timeline that the work is intended to follow. All updates/modifications to the proposed timeline will be posted at <http://www.cs.rit.edu/~sls6076/Xpointproject/>

Date	Proposed Work	Status
September 26 th 2005	Pre-proposal	Completed
Phase1 (Setting up)		
September 30 th 2005	Develop website to maintain project status	Completed
October 2nd 2005	Determine software, parsers needed for the project. Set up working environment for the project (xpoint set and run examples), Redo proposal changes and determine project team members.	Completed
Phase 2 (Learning Curve)		
October 15 th 2005	Workable knowledge for XSLT, XPath, DTD examples, Xpoint	Completed
November 15 th 2005	XSL-FO development, examples.	Completed
Phase 3 (Development Work)		
November 30 th 2005	WYSIWYG editor for xpoint presentation xml data.	Completed
December 15 th 2005	Generate sample PDF document for xml data	Completed
Jan 15 th 2006	Generate sample html slides using cocoon	Completed
Jan 19 th 2006	Submit Project Proposal	Completed
Feb 15 th 2006	Develop form/xml schema	Completed
Feb 28 th 2006	Develop multiple html slides from xpoint xml	Completed
Mar 02 nd , 2006	Modified CSS to get multiple output	Completed
March 17 th 2006	Generated PDF document using XSL-Fo	Completed
March 25 th 2006	Configure the entire project under one umbrella	Completed
April 30 th 2006	Configure JEdit to the Xpoint environment	
Phase 5 (Documentation)		
May 10th 2006	Develop Technical Report and study portability issues.	
May 26 th 2006	Proposed Defense Date	

7. References

1. X-Point by Dr. Schreiner
<http://www.cs.rit.edu/~ats/projects/xpoint/index.html>
2. Dr. Schreiner's website <http://www.cs.rit.edu/~ats/>
3. XSLT: Programmer's Reference, 2nd Edition by Michael Kay
4. XSLT by Doug Tidwell, First Edition August 2001
5. XSL Tutorial by W3 schools
http://www.w3schools.com/xsl/xsl_languages.asp
6. XML Path Language (XPath) by James Clark, Steve DeRose
7. Open Office Suite (Stable version)
<http://download.openoffice.org/1.1.4/index.html>
8. http://www.w3schools.com/xsl/xsl_transformation.asp
9. Cocoon: <http://cocoon.apache.org/>
10. Eclipse: <http://www.eclipse.org/>
11. Altova Authentic: http://www.altova.com/products_doc.html
12. XSL-FO Tutorial: http://www.w3schools.com/xslfo/xslfo_reference.asp
13. XML Schema: <http://www.xml-training-guide.com/xml-schema.html>
14. JEdit: <http://www.jedit.org/>

8. Appendix

I. Appendix A

```
- <xpoint>
  <title>XPoint </title>
+ <body></body>
+ <section></section>
- <section>
  <title>DTD </title>
  + <body></body>
  + <topic></topic>
  + <topic></topic>
  + <topic></topic>
  </section>
+ <section></section>
- <information>
  + <author></author>
  + <author></author>
  <date>09 Dec 2005 </date>
  <time> 12:45 pm </time>
  </information>
</xpoint>
```

Description:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xmlspysps xpoint.sps?>
<xpoint>
  <title>
    XPoint
  </title>
  <body>
    <bold>XPoint</bold> is a simple <italic>presentation
    graphics </italic> system. In which presentation is
    specified in <link>XML</link> and converted to HTML for
    browsing. <underline> Appearance </underline> is controlled
    using CSS.
  </body>
  <section>
    <title>
      Text
    </title>
    <body>
      A presentation can be created as a text table and
      converted with <bold>awk </bold> into XPoint format
    </body>
    <topic>
      <title>
        Markup
```

```
</title>
<body>
  Paragraph styles XPoint, Section and Topic
  introduce structure and titles, Body can be
  used to subdivide a topic.
  Character styles File and CSS allow to
  prescribe output and style sheet file name
  stems. Style can be used to define properties
  to be inherited by the page body.
  Paragraph style P identifies running text, Pre
  marks verbatim code segments.
  Character styles B, I, and TT indicate emphasis
  and code references.
</body>
</topic>
</section>
<section>
  <title>
    DTD
  </title>
  <body>
    A presentation must conform to a <italic>Document
    Type Definition.</italic>XHTML markup is permitted.
  </body>
  <topic>
    <title>
      Elements
    </title>
    <body>
      xpoint contains the title page and the
      sections.<b>Each section contains a title
      page and some topics.</b> Each topic has a
      title and one or more pages.
    </body>
  </topic>
  <topic>
    <title>
      Content
    </title>
    <body>
      title and body can <src> C:\cocoon-
      2.1.7\build\webapp\html-pdf\image01.jpg</src>
      contain <b> arbitrary XHTML markup.</b>
    </body>
  </topic>
  <topic>
    <title>
      Links
    </title>
    <body>
      body can prescribe the stem of the style sheet
      and output file name and add properties to the
      style sheet.The file name stem should not start
      with an underscore.The generated file name is
      slides<italic>/file.html </italic>
    </body>
  </topic>
</section>
</body>
</html>
```

```
        </topic>
</section>
<information>
  <author>
    <name>
      Sudhanshu Sood
    </name>
    <contact>
      12501 Laurel Grove Pl. Germantown, MD 20874
    </contact>
    <email>
      ansh78@yahoo.com
    </email>
  </author>
</information>
</xpoint>
```

II. Appendix B

File: Describes the XML Schema to validate Xpoint document.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="xpoint">
    <xs:annotation>
      <xs:documentation>
        Comment describing your root element
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="information"/>
        <xs:element ref="title"/>
        <xs:element ref="body"/>
        <xs:element ref="section" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="author">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name"/>
        <xs:element name="contact"/>
        <xs:element name="email"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="body" type="formatType"/>
  <xs:element name="section">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="body" type="formatType"/>
        <xs:element ref="topic" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="topic">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="title" type="xs:string"/>

```

```
        <xs:element name="body" type="formatType"
            maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:complexType name="formatType" mixed="true">
    <xs:choice minOccurs="0" maxOccurs="unbounded">
        <xs:element name="bold" type="formatType"/>
        <xs:element name="italic" type="formatType"/>
        <xs:element name="underline" type="formatType"/>
        <xs:element name="link" type="formatType"/>
    </xs:choice>
</xs:complexType>
<xs:element name="information">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="author" maxOccurs="unbounded"/>
            <xs:element name="date" type="xs:date"/>
            <xs:element name="time" type="xs:dateTime"/>
            <xs:element name="location"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema
```

III. Appendix C

File: Form to describe presentation. Source of the form is an xml document which is validated by the corresponding XML Schema.

Body	Title	Xpoint
	CSS	add css
	File	add file
Style	add style	
Section	Title	DTD
	CSS	add css
	File	add file
	Style	add style
	Title	Elements
	CSS	add css
	File	add file
	Style	add style
	Title	Content
	CSS	add css
	File	add file
	Style	add style
Title	File Structure	
CSS	add css	
File	add file	
Style	add style	

IV. Appendix D

Source file: **sitemap.xmlmap**

```
<!--Configuration file used to generate navigable html for Xpoint xml
document. @Sudhanshu Sood -->
<?xml version="1.0" encoding="iso-8859-1"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
<!-- define the Cocoon processing pipelines -->
<map:pipelines>
  <map:pipeline>
    <!-- to process CSS file specified for the presentation □
    <map:match pattern="css/*">
      <map:read src="css/{1}" mime-type="text/css"/>
    </map:match>
  </map:pipeline>
  <map:pipeline>
    <!--browser matched presentation to output xsl □
    <map:match pattern="presentation">
      <map:generate src="xml/xpoint.xml"/>
      <map:transform src="xslt/xpointsection.xsl"/>
      <map:transform src="xslt/section2html.xsl"/>
      <map:serialize type="xhtml"/>
    </map:match>
    <!--matches all the sections specified-->
    <map:match pattern="section/*">
      <map:generate src="xml/xpoint.xml"/>
      <map:transform src="xslt/xpointtopic.xsl">
        <map:parameter name="section" value="{1}"/>
      </map:transform>
      <map:transform src="xslt/sectionhtml.xsl"/>
      <map:serialize type="xhtml"/>
    </map:match>
    <!--matches all the sections specified in a section-->
    <map:match pattern="section/*/topic/*">
      <map:generate src="xml/xpoint.xml"/>
      <map:transform src="xslt/topicbody.xsl">
        <map:parameter name="section" value="{1}"/>
        <map:parameter name="topic" value="{2}"/>
      </map:transform>
      <map:transform src="xslt/sectionhtml.xsl"/>
      <map:serialize type="xhtml"/>
    </map:match>
  </map:pipeline>
</map:pipelines>
</map:sitemap>
```

V. Appendix E

Sample XSL-FO file used to generate a PDF document for the xpoint format.

File name: Xpoint2pdf

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0"
  xmlns:fo="http://www.w3.org/1999/XSL/Format"
>
<!-- generate PDF page structure -->
<xsl:template match="/">
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="intropage"
      page-height="29.7cm"
      page-width="21cm"
      margin-top="1in"
      margin-bottom="2in"
      margin-left="2.5cm"
      margin-right="2.5cm"
    >
    <fo:region-body
      padding-start="1cm"
      padding-end="1cm"
      margin-top=".6in"
      margin-bottom=".6in"
      margin-left="0.7in"
      margin-right=".5in" />
    </fo:simple-page-master>
    <fo:simple-page-master master-name="contents"
      page-height="29.7cm"
      page-width="21cm"
      margin-top="1in"
      margin-bottom="1.5in"
      margin-left="2.5cm"
      margin-right="2.5cm"
    >
    <fo:region-body
      padding-start="1cm"
      padding-end="1cm"
      margin-top=".6in"
      margin-bottom=".6in"
      margin-left="0.7in"
      margin-right=".5in" />
    <fo:region-before extent="0.5in" />
    <fo:region-after extent="0.5in" />
    </fo:simple-page-master>
    <fo:simple-page-master master-name="desc"
      page-height="29.7cm"
      page-width="21cm"
      margin-top="1in"
      margin-bottom="0.7in"
      margin-left="1.5cm"
      margin-right="2.5cm"
    >
```

```
>
<fo:region-body
  padding-start="1mm"
  padding-end="3mm"
  margin-top="1in"
  margin-bottom=".5in"
  margin-left="15mm"
  margin-right="15mm" />
<fo:region-before extent="0.7in" />
<fo:region-after extent="0.4in" />
</fo:simple-page-master>
</fo:layout-master-set>
<fo:page-sequence master-reference="intropage">
  <fo:flow flow-name="xsl-region-body">
    <xsl:call-template name="IntroPage" />
  </fo:flow>
</fo:page-sequence>
<fo:page-sequence master-reference="contents" >
  <fo:flow flow-name="xsl-region-body">
    <xsl:apply-templates select="/" mode="toc" />
  </fo:flow>
</fo:page-sequence>
<fo:page-sequence master-reference="desc" >
  <fo:flow flow-name="xsl-region-body">
    <xsl:call-template name="presentation" />
  </fo:flow>
</fo:page-sequence>
</fo:root>
</xsl:template>
<xsl:template name="IntroPage">
<fo:block>
<fo:block text-align="center"
  font-size='26pt'
  space-before = '20mm'
  space-after = '25mm' >
  <xsl:value-of select='/xpoint/title' />
</fo:block>
<fo:block text-align="center"
  font-size='18pt'
  space-before = '25mm'
  space-after = '12mm'>by
</fo:block>
<fo:block text-align="center"
  font-size='18pt'
  space-before = '25mm'
  space-after = '12mm'>
<xsl:for-each select="/xpoint/information/author">
  <xsl:value-of select="name" />,
  <xsl:value-of select="email" />
</xsl:for-each>
</fo:block>
</fo:block>
<fo:block text-align='center' font-size='10pt'
  space-before = '15mm'>
  Date: <xsl:value-of
  select="/xpoint/information/date" />
```

```
</fo:block>
<fo:block text-align='center' font-size='10pt'>
    Time:<xsl:value-of select="/xpoint/information/time"/>
</fo:block>
</xsl:template>
<xsl:template name="presentation">
    <fo:block>
        <fo:block text-align="center" font-weight="bold" >
            <xsl:value-of select="/xpoint/title"/>
        </fo:block>
        <fo:block text-align="justify">
            <xsl:value-of select="/xpoint/body"/>
        </fo:block>
        <xsl:apply-templates select="xpoint/section"/>
    </fo:block>
</xsl:template>
<xsl:template match="section">
    <fo:block space-before="1.5mm">
        <fo:block font-weight="bold">
            <xsl:value-of select="title"/>
        </fo:block>
        <fo:block text-align="justify">
            <xsl:value-of select="body"/>
        </fo:block>
        <xsl:apply-templates select="topic"/>
    </fo:block>
</xsl:template>
<xsl:template match="topic">
    <fo:block margin-left=".75cm">
        <fo:block font-weight="bold">
            <xsl:value-of select="title"/>
        </fo:block>
        <fo:block text-align="justify" font-size="10"
            margin-left=".75cm">
            <xsl:value-of select="body"/>
        </fo:block>
    </fo:block>
<xsl:element name="fo:external-graphic">
    <xsl:attribute name="src"><xsl:value-of select="src"/>
</xsl:attribute>
</xsl:element>
</fo:block>
</fo:block>
</xsl:template>
<xsl:template match="/" mode="toc">
    <fo:block >
        <fo:block text-align="center" font-size="16pt" spaceafter="
            10mm">
        </fo:block>
        <xsl:apply-templates select="/xpoint/section" mode="toc"/>
    </fo:block>
</xsl:template>

<xsl:template match="section" mode="toc">
    <fo:block>
        <fo:inline>
```

```
        <xsl:value-of select="title"/>
        <xsl:apply-templates select="topic" mode="toc"/>
    </fo:inline>
</fo:block>
</xsl:template>
<xsl:template match="topic" mode="toc">
    <fo:block font-size="12pt" text-align-last="justify"
        margin-left="1.0cm">
        <fo:inline>
        <xsl:value-of select="title"/>
        <fo:leader leader-pattern="dots"/>
        </fo:inline>
    </fo:block>
</xsl:template>
</xsl:stylesheet>
```