

Chapter 3

Background: Related technologies

The ISO 19115 Community Adaptor will be implemented as a web service on a Windows web server. It will utilize some of the newest Internet technologies such as, web service, XML, and XSLT. For development, it will be developed using Microsoft Visual Studio with ASP.Net and C# support. The Microsoft SQL Server will be used for database support. It will shorten the learning curve to keep them all within the Windows platform. Here are some backgrounds about those related technologies.

3.1 Web Service

Web service is a popular Internet technology. It is a low cost communication method using a standardized protocol to achieve application interoperability. It is an interface between application code and the user. Interoperability is one of the key benefit of Web Service.

3.1.1 What is web service?

A web service is a communication method between two electronic devices over the Internet. The W3C definition of the web service: "a software system designed to support interoperable machine-to-machine interaction over a network." Web service uses WSDL (Web Service Description Language) to describe the web service. The WSDL is in a machine-processable format.

3.1.2 Classes of Web Service

According to W3C, there are 2 major classes of web services. REST-compliant web service and arbitrary web service. RESTful web services' primary purpose is the manipulation of XML representations of web resources using a uniform set of "stateless" operations while the arbitrary web services may expose an arbitrary set of operations.

3.1.3 Web service architecture

The following illustration (Figure 3.1) depicts the web service architecture.

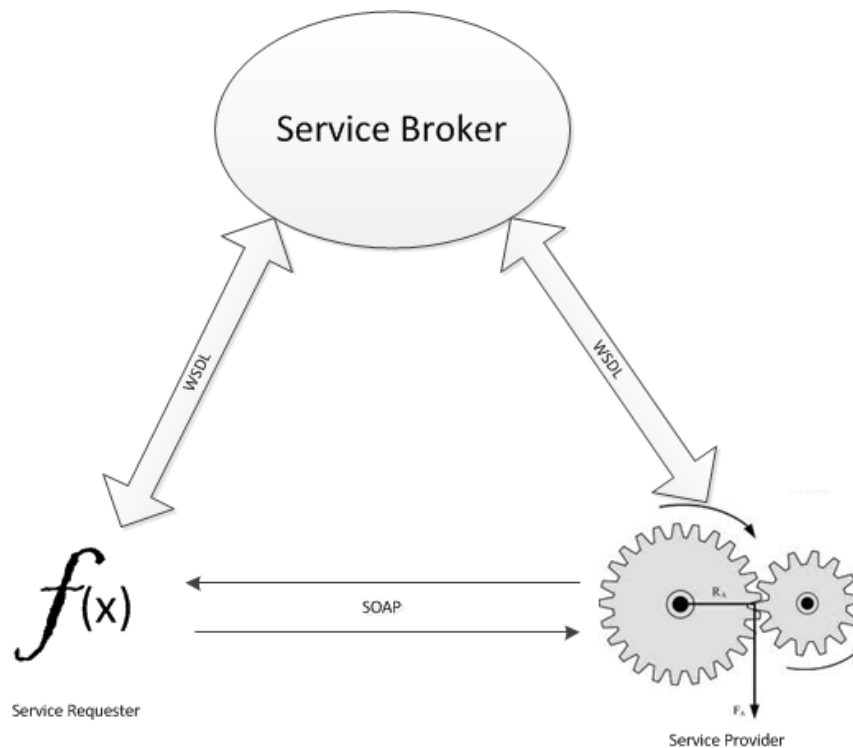


Figure 3.1: Web service architecture: Web services provide a standard means of interoperating between different software applications, running on a variety of platforms and/or frameworks

Discovery
Description
Packaging
Transport
Network

Table 3.1: The web service Technology Stack

3.1.4 The web service Technology Stack

The web service architecture is implemented through layering of five types of technologies. They are organized into layers and build upon one another. See Table 3.1

3.2 XML

3.2.1 What is XML?

Extensible Markup Language (XML) is an open standard markup language that defines a set of rules for encoding documents in a format that is readable by both humans and machines.

XML is extensible. You can make up your own tags and uses tag-based syntax. It is the foundation for the next generation web technologies like XHTML, RSS, Ajax, and web service. It is not the replacement for HTML.

3.2.2 What does XML Do?

XML is used to structure and describe information. It provides a way to exchange information between different systems over the Internet.

XML is designed to store, retrieve and transport data. XML files are text files so it is platform-independent. That makes it a perfect choice for data transportation between machines or over the Internet. It is meant for small file rather than large files. Many software applications use XML for preference settings, look-up tables, work files etc. For example, a calendar application can use XML file to store holidays instead of hard-coding them inside the source code. The end users can add new, personal

holidays, change or delete existing holidays by simply editing the holiday XML file after the application is compiled. The holiday can be categorized using the XML entity attributes and the application can decide which holidays to include in a calendar.

XML only describes the data. It does not include presentational instructions. There are a number of ways for formatting a XML file. The easiest way is to use CSS (cascading style sheet). CSS works the same way as it does for HTML. The XML file includes a reference to a CSS file. A CSS compliant browser will format the XML according to the instruction stored in the CSS file. CSS is the simplest way of formatting XML. It works like find & replace.

XSLT technology provides a powerful and more flexible and versatile technique to format XML files. XSLT transformation is a process that breaks apart a XML file and rebuilds a completely different file format. The XSLT processor takes a XML file and XSL (eXtensible Stylesheet Language) as inputs and transform the XML file according to the instruction provided in the XSL file and produces a new output file. (see figure xx). Together with XPath and Xpointer, XSLT provides more powerful programming features like search, sort, aggregation (calculation) and so on. XSLT is heavily used in this project.

While XML is very useful, there are some drawbacks as well. While XML offers great power of flexibility, platform independent, and easy to transport, it must ensure data integrity. XML files can be corrupted, truncated, incomplete, or mistyped and a corrupted data file can cause unexpected consequences. XML offers different mechanisms to ensure XML document integrity. The first technique is to check for well-formedness. It basically checks for missing tags, malformed tags, illegal characters, and so on. A XML document must be well-formed, meaning that the document adheres to the syntax rules specified by the XML 1.0 specification. Well-formedness requirements include: content be defined, content be defined with a beginning and end tags, and content be properly nested (each XML document must contain one and only one root element, parents within root and children within parents) and so on. The well-formedness is ensured by XML parsers. The XML parser must report any problems to users.

A well-formed XML only guarantees that it is grammatically correct, not necessarily a valid XML document. Most applications require the validation test. A valid XML document is a well-formed XML document that also passes the validation test. For a validation test, XML authors supply a DTD or schema for the XML document. The XML parser validates the XML file against the DTD or schema.

The validation rules can be embedded inside the XML document or stored in an external file and provide the reference link inside the XML. Example of the validation rules include: what entities are allowed, number of children, allowed values for a particular entity.

3.2.3 XML-related technologies

As mentioned above, XML is the foundation of the next generation of the Internet. XML provides a way to structure information. To present, search, and modify the information on the web, other related technologies have been developed. Here is a list of some of the related technologies:

- Xpath – Extensive path language. It uses path-like syntax.
- XSLT – styling language
- XQuery – used to query XML data like SQL
- Xpointer and Xlink – used to create hyperlinks to XML Documents, and arbitrary points within XML documents

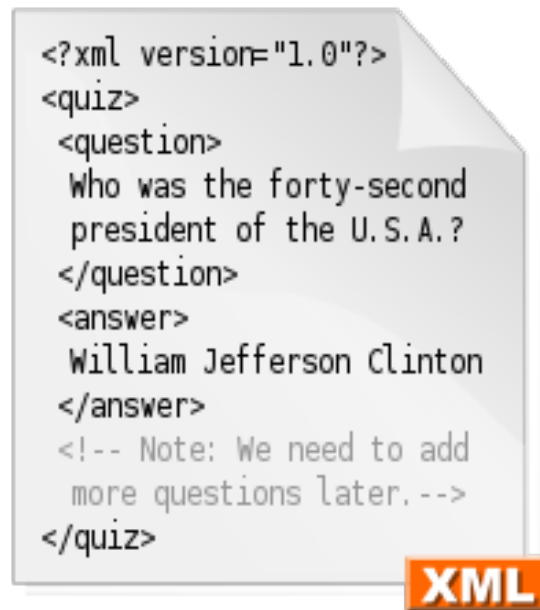


Figure 3.2: XML: a sample XML file

3.3 XSLT

3.3.1 What is XSLT?

As I mentioned before, XML is a popular, powerful, and flexible mark up language to store and retrieve data. The flexibility gives us the power to develop our own vocabularies. This presents the need to be able to transform documents marked up in one XML vocabulary to another.

The XSLT (eXtensible Stylesheet Language Transformation), along with XPath (XML Path Language), meets that need. The XSLT provides a powerful implementation of tree-oriented transforma-

tion language that transfers XML instance using one vocabulary into either another XML document in a different vocabulary or a non-XML document like text file. XSL (eXtensible Stylesheet Language) is used to specify the instructions for a XSLT processor to generate the desired output from the given input XML document.

XSLT enables and empowers interoperability among XMLs in different domain.

3.3.2 Logical components of an XSLT application

The XSLT transformation is carried out by a XSLT processing software (XSLT Engine). The XSLT Engine takes one XML document and XSL document as inputs and generate a desired output file(s) according to the instructions provided within the XSL input document. see Figure 3.3

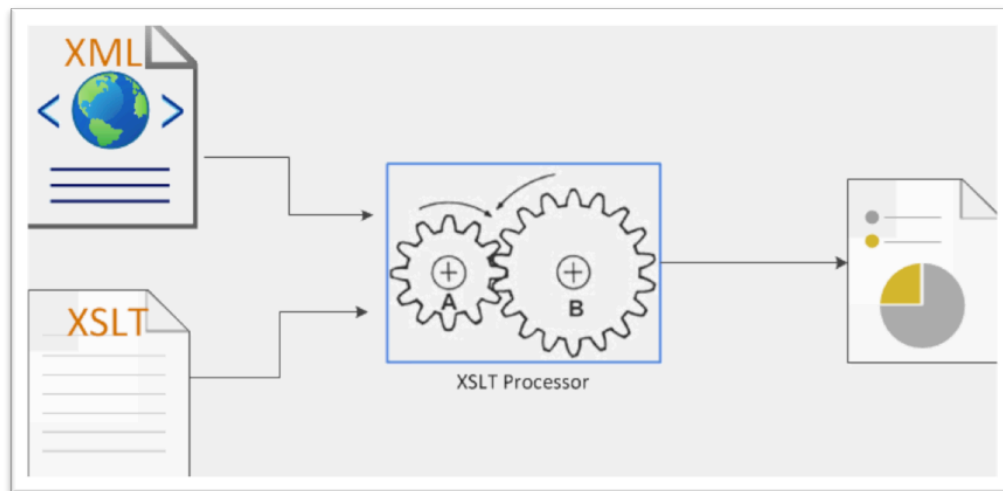


Figure 3.3: XSLT processor takes one or more XML documents and one or more XSLT files as input and produce a new XML document

The first component is the input XML file. The input XML document is a text file that was made of a sequence of data characters and markup tags. Data elements are delimited with start-tags and end-tags. The XSLT engine does not work on the XML input file directly. It builds a tree based on the XML input file and works on the tree.

The second component is the XSLT stylesheet (XSL file). A XSL file is a well formed XML document. It acts like a computer program that includes all the transformation instructions. It is called a stylesheet file but it is totally different from the HTML stylesheet (CSS) file. The XSL file does not have “style” information but instead “transformation” information, and thus should be called a transformation file. The XSL file contains a series of template rules. Each rule is a sequence of XSLT commands and each command is a XML element with attributes. A rule is executed when it matches some conditions or is called by name.

Another component of a XSLT application is the XSLT engine. There exists several ways to perform a transformation. For stand-alone XSLT engines, users download the application (many free open source XSLT processors) and execute a transformation using a command line like:

```
Qiping > Saxon [input XML] [input XSL] [output file]
```

Popular XSLT processors include:

- Saxon (<http://users.iclway.co.uk/mhkay/saxon/>)
- C Xalan XSLT (<http://xml.apache.org/xalan/index.html>)
- C Unicorn XSLT Processor (<http://www.unicorn-enterprises.com/>)
- C XSLT C library for Gnome (<http://xmlsoft.org/XSLT/>)

In a web environment, an XSLT transformation can occur in the web server or on the client side. All the major web browsers (recent versions) support XSLT. Or inside a webpage, Javascript can be used to open and transform a XML file.

All major browsers have support for XML and XSLT.

Mozilla Firefox

Firefox supports XML, XSLT, and XPath from version 3.

Internet Explorer

Internet Explorer supports XML, XSLT, and XPath from version 6. Internet Explorer 5 is NOT compatible with the official W₃C XSL Recommendation.

Google Chrome

Chrome supports XML, XSLT, and XPath from version 1.

Opera

Opera supports XML, XSLT, and XPath from version 9. Opera 8 supports only XML + CSS.

Apple Safari

Safari supports XML and XSLT from version 3.

Another way to transform XML is in the server using frameworks like ASP.NET. Here is a sample transformation in ASP: ASP XSLT Transformation

```

1 <%
  'Load XML
3 set xml = Server.CreateObject("Microsoft.XMLDOM")
  xml.async = false
5 xml.load(Server.MapPath("cdcatalog.xml"))

7 'Load XSL
  set xsl = Server.CreateObject("Microsoft.XMLDOM")
9 xsl.async = false
  xsl.load(Server.MapPath("cdcatalog.xsl"))
11
12 'Transform file
13 Response.Write(xml.transformNode(xsl))
%>

```

Javascript implementation

```

<html>
2 <head>
  <script >
4
5   function displayResult ()
6   {
7     xml=loadXMLDoc("cdcatalog.xml");
8     xsl=loadXMLDoc("cdcatalog.xsl");
9     // code for IE
10    if (window.ActiveXObject)
11    {
12      ex=xml.transformNode(xsl);
13      document.getElementById("example").innerHTML=ex;
14    }
15    // code for Mozilla , Firefox , Opera , etc .

```



```

16 else if (document.implementation && document.implementation.createDocument)
    {
18     xsltProcessor=new XSLTProcessor();
    xsltProcessor.importStylesheet(xsl);
20     resultDocument = xsltProcessor.transformToFragment(xml,document);
    document.getElementById("example").appendChild(resultDocument);
22     }
    }
24 </script>
</head>
26 <body onload="displayResult()">
<div id="example" />
28 </body>
</html>

```

XML document: cdcatalog.xml

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
3 <catalog>
    <cd>
5         <title>Empire Burlesque</title>
        <artist>Bob Dylan</artist>
7         <country>USA</country>
        <company>Columbia</company>
9         <price>10.90</price>
        <year>1985</year>
11    </cd>
    .
13 </catalog>

```

XSL document: cdcatalog.xsl

```

2 <?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
4 xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

6 <xsl:template match="/">
    <html>
8    <body>
        <h2>My CD Collection</h2>

```

```
10 <table border="1">
    <tr bgcolor="#9acd32">
12       <th align="left">Title </th>
        <th align="left">Artist </th>
14     </tr>
    <xsl:for-each select="catalog/cd">
16     <tr>
        <td><xsl:value-of select="title" /></td>
18        <td><xsl:value-of select="artist" /></td>
        </tr>
20    </xsl:for-each>
    </table>
22 </body>
    </html>
24 </xsl:template>
    </xsl:stylesheet>
```

The last component of the XSLT application is the output file. A XSLT transformation can produce one or more output files in format of XML, HTML, or text (untagged files).

XSLT is a popular technology. Many businesses are using it in some degree because it is easy, cheap, faster. However it is not good at everything. It cannot convert something else into a XML file. It is not good with non-XML data like Word document. It is not as good as programming languages for computation. Also it is not suited for large data files. It is weak on string manipulation.

3.4 ASP.NET/C#

3.4.1 What is ASP.NET

ASP.NET is a server-side web application development framework from Microsoft. It is a popular framework on the Windows platform to build dynamic websites, web applications, and web services. Microsoft released the 1.0 .NET framework in January 2002. It is the successor of Microsoft's Active Server Pages (ASP) technology. ASP.NET is build on Common Language Runtime (CLR) to allow programmers to write ASP.NET code in any supported languages like Visual Basic, C#. One of the fundamental advances in ASP.NET over ASP is the separation of presentation from control. With ASP, a dynamic web page is achieved by including control code (Visual Basic, Javascript) inside the HTML

page. For ASP.NET, Microsoft introduced the “code-behind” model to separate a webpage’s formatting from its logical control. Each webpage (.aspx) has a corresponding “code-behind” page (.aspx.cs) to handle programming logic to control the page behavior.

3.4.2 ASP.NET Fundamentals

ASP.NET introduced an entirely new object-oriented execution model. ASP.NET execution centers around CLR classes that implement an interface named `IHttpHandler`.

When it arrives on an IIS webserver, the HTTP request/response is routed through many server side objects for processing. When a user hits the Return key after entering a URL, the web browser sends a GET request to the target web server. The request travels through various routers and reaches the web server on port 80. The IIS on the server will handle the request. The IIS intercepts the request and maps it to the worker process and the request follows a very specific path through the pipeline. The path is different depending on the IIS version. For IIS 7.X, it works as follows:

1. The browser makes a request for a resource on the Web server .
2. HTTP .SYS picks up the request on the server .
3. HTTP .SYS uses the WAS to find configuration information to pass on to the WWW Service .
4. WAS passes the configuration information to the WWW Service, which configures HTTP .SYS .
5. WAS starts a worker process in the application pool for which the request was destined.
6. The worker process processes the request and returns the response to HTTP.SYS .
7. HTTP .SYS sends the response to the client .

3.4.3 Event-Driven Application

As we know, most of the modern popular web development platforms, like PHP, ASP, and Cold Fusion, support the OOP (Object Oriented Programming) concept. OOP is effective in keeping web application code well organized, modular, and scalable, especially for large, complex projects.

EDP (Event Driven Programming), on the other hand, is less common among web development

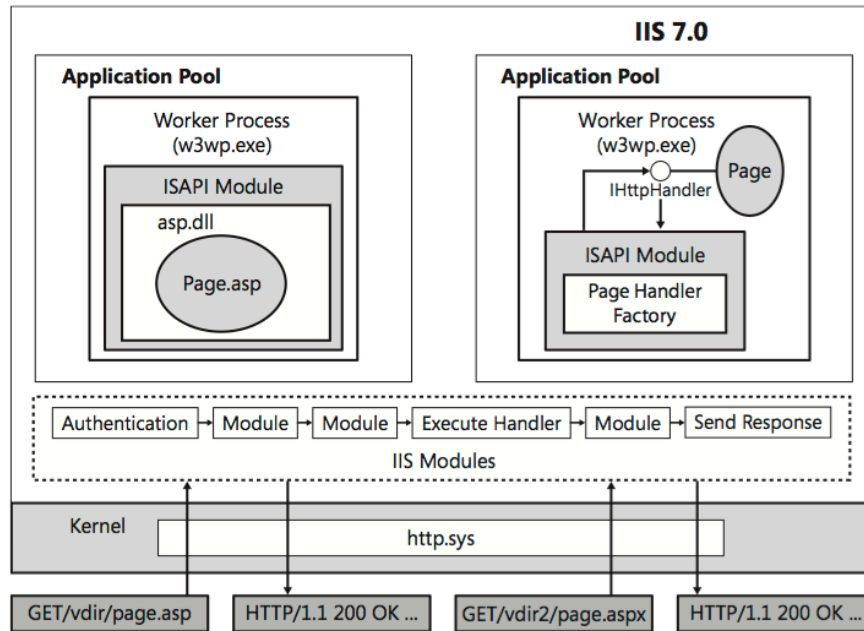


Figure 3.4: ASP.NET and IIS 7.x architecture

platforms. ASP.NET is the one that supports both OOP and EDP. It provides an unmatched level of developer productivity on complex projects.

Before EDP, most programs are procedural. A procedural program is like a list of instructions. Computer execute those instructions stored in a program one by one with some conditional branches upon user input or the result of previous action. Procedural program works fine for simple console applications. But it is not enough for the modern GUI (Graphical User Interface). Instead of executing instructions, an EDP program do not do anything unless “something” happened. That “something” is called events. An event driven program responds to events. Events can be generated or fired by a user action or by system (like file is loaded). In ASP.NET, objects may raise events and may have assigned event handlers. The event handlers respond to events by executing the code inside the event handler. ASP.NET names event handlers by prepending the work “On” to the event name. For example, “OnClick” for Click event.

Table 3.2 shows a list of common events and event handler names:

3.4.4 ASP.NET Server Controls

In order for the ASP.NET code behind to communicate with the HTML controls in an ASP.NET page, the controls must be enclosed inside the <form> tag. ASP.NET <form> tag is different from the regular HTML <form> tag. Any HTML controls can be placed inside the ASP.NET <form> tag. It is like a regular HTML web page for the web browser to format and display the page. In order for the behind code to retrieve and set the controls' attributes or value, the "RunAt" attribute must be set to "Server". Then the code behind can access the controls by referencing to controls by ID.

3.4.5 ASP.NET Sessions

3.4.6 Web Service Support in ASP.NET

As I mentioned in the Web Service section, web service includes providing web service and consuming web service. By providing web service, I mean that when a user send a web service request to a web site or a web application in a web server, the web application will process the request, (for example, fetch data from the database, invoke another application, do the calculation, format the result and send the result back to user). By consuming web service, we mean that we construct a web service request, manually or automatically by a program, send the request to a different web server or to the same web server, or maybe in the same website. When the reply arrives the application process or parses the SOAP message. With the information the application format it as a new web service response or as a web page and

Event name	Event handler name	Applies to
BubbleEvent	OnBubbleEvent	All controls
CheckedChanged	OnCheckedChanged	CheckBox
Click	OnClick	Button, LinkButton, ImageButton
DataBinding	OnDataBinding	All controls
Init	OnInit	All controls
ItemCreated	OnItemCreated	Repeater
ItemDataBound	OnItemDataBound	Repeater
Load	OnLoad	All controls
PreRender	OnPreRender	All controls
SelectedIndexChanged	OnSelectedIndexChanged	DataGrid, DataList, CheckBoxLayout, DropDownList, ListBox, RadioButtonList
TextChanged	OnTextChanged	TextBox
Unload	OnUnload	All controls

Table 3.2: Common events and event handler

send to the user. In my opinion, it is logically easier to provide web service than consume web service since we have to deal with various message structures when consuming web services.

Web service is simple and easy to understand. It is an application what can send data as XML conforming to the SOAP specification. It is straightforward to build a web application capable of receiving SOAP messages over the HTTP and delivering SOAP messages out of it. The ASP.NET just make it easier for web application developers to provide/use web service by providing the framework to do the plumbing work. The developers can focus on the application logic, instead of figuring out how HTTP, SOAP WSDL or any of the technologies that form the basis for web service. The Microsoft Visual Studio .NET provide a rich feature set for programmers to easily develop well-documented scalable web service application.

3.5 SQL Server

Microsoft SQL Server is a RDMS (Relational Database Management System) developed by Microsoft. It is one of the popular RDMS in the market. (Other popular RDMS's include Oracle, IBM's DB2, and MySQL) The newest version is SQL Server 2012 that is cloud-ready. Microsoft SQL Server is used by small, medium, and large enterprises. One of its advantages is that the MS SQL Server encompasses more than just a relational database engine, it includes business intelligence, a complex event processing engine, highly scalable data warehousing solutions, and a SQL Server edition running in the cloud.

Microsoft offers different editions for different clients. SQL Server editions include: Enterprise, Business Intelligence, Standard, Web, Express, Compact, and SQL Azure. All the SQL Server editions can only run in a Microsoft Windows system. The Enterprise, Business Intelligence, and Web editions only can be installed on a Windows Server platform while rest of the editions can also be installed in Windows systems like Windows 8, 7, and Vista.

Microsoft breaks down its SQL Server editions in almost the same way as Oracle, with Express, Standard, and Enterprise editions being the main products. But Microsoft does not offer additional options that can be purchased separately as Oracle does, such as Partitioning, Spatial, OLAP, and Data Mining. Instead, it bundles all its features into the base product. The available features depend on the editions purchased.

3.6 ISO 19115

3.6.1 Metadata

As mentioned in last chapter, metadata is commonly defined as “data about data”, “structured data about data”, “data which describes attributes of a resource”, or even “information about data”. Metadata provides information about one or more aspects of the data, such as purpose of the data, means of creation of the data, owner or author of the data, point of contact of the data, etc.

The ISO 19115 is a metadata standard developed and administrated by the International Organization for Standardization. It is designed to describe geographic datasets. The ISO 19115, which is part of the ISO 19000 series, has two versions: ISO 19115:2003 and ISO 19115:2009. The ISO 19115:2009 added support for imagery and gridded data.

Compared to other geographic metadata standard like FGDC and DC, the ISO 19115 is a new standard. But it is gaining popularity and being adopted by many countries and organizations. The ISO 19115 is the main metadata standard in the implementation of this project.

3.6.2 Metadata standard

As described in [4], “metadata standards are requirements which are intended to establish a common understanding of the meaning or semantics of the data, to ensure correct and proper use and interpretation of the data by its owners and users.” A metadata standard is usually established by national and international standard communities like ANSI (American National Standards Institute) and ISO (International Organization for Standard).

3.6.3 Geographical Metadata Standards

In the geographic domain, the most common metadata standards are FGDCs Content Standard for Digital Geospatial Metadata (CSDGM) and the recently ratified ISO 19115 [12]. The CSDGM standard contains over 300 data and compound elements while the ISO 19115 has over 400 elements (divided into 14 metadata packages) in 86 classes that have 282 attributes and 56 relations. The ISO 19115 was developed by the geospatial community to address specific issues relating to both the description

and the curation of spatial data[5]. The ISO 19115s abstract models are written using the UML (Unified Modeling Language). The accompanying XML schema, ISO/CD TS 19139, enables interoperable XML expression of ISO 19115 compliant metadata.