SEMANTIC WEB SERVICES ENABLED CMS

SOH JIEN MIN

MASTER OF COMPUTER SCIENCE

FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN
MARCH 2014

SEMANTIC WEB SERVICES ENABLED CMS

By

SOH JIEN MIN

A project submitted to the Department of Internet Engineering and Computer Science,
Faculty of Engineering and Science,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Master of Computer Science
March 2014

TABLE OF CONTENTS

ABSTRACT

SEMANTIC WEB SERVICES ENABLED CMS

Soh Jien Min

The growing number of information and contents in the Internet has made search engines far more capable of locating results that are relevant and accurate. The major search engines that have already implemented semantic search engines to help getting more relevant search than what normally traditional text search can't do. Instead of depending on the search engines to do the semantic process on the static content, which will limit the quality of the search result, the content structure is the key benefits of what the search engine can get. To get to structure the content, the Content Management System (CMS) will be the most suitable place to start with. We will conduct a research on developing an interface between the conventional CMS method of entering contents, and publishing these contents into structure that in turns helps machines or computers to process the search more relevant, accurate and meaningful. The interface should require minimal technical knowledge to operate and should be as streamline as possible to retain simplicity of the CMS.

ACKNOWLEDGEMENTS

Firstly I want to thank Prof. Dr. Victor Tan Hock Kim, my supervisor for this project. For his support and knowledge in the field, as well as encouragement and guidance in helping complete the project. I appreciate his patience and understanding towards my diverted attention due to my busy timetable and will always give advices as soon as he can. Without him it would be unimaginable that I could complete this project.

I would also like to thank DCLabs MSC Sdn. Bhd. for giving me permission to use source code and the in house product CMS for research and development purposes. As well as giving me flexible work time so I can complete the project.

Soh Jien Min

FACULTY OF ENGINEERING AND SCIENCE
UNIVERSITI TUNKU ABDUL RAHMAN

Date: 28 MARCH 2014

PERMISSION SHEET

It is hereby certified that Soh Jien Min (ID No:09UEM08881) has completed this final year project entitled "SEMANTIC WEB SERVICES ENABLED CMS" under the supervision of Prof. Dr. Victor Tan Hock Kim (Supervisor) from the Department of Internet Engineering And Computer Science, Faculty of Engineering and Science, and _____ (Co-Supervisor)* from the Department of _____, Faculty of Engineering and Science.

I hereby give permission to the University to upload softcopy of my final year project in PDF format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

_____
SOH JIEN MIN

APPROVAL SHEET

This dissertation/thesis entitled "SEMANTIC WEB SERVICES ENABLED CMS" was prepared by Soh Jien Min and submitted as partial fulfilment of the requirements for the degree of Master of Computer Science at Universiti Tunku Abdul Rahman.

Approved by:

_____
(Prof. Dr. VICTOR TAN HOCK KIM)
Date:……..
Professor/Supervisor
Department of Internet Engineering and Computer Science
Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

_____
(Prof. Dr.)
Date:……..
Professor/Co-supervisor
Department of _____
Faculty of _____
Universiti Tunku Abdul Rahman

DECLARATION

I Soh Jien Min hereby declare that the dissertation is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTAR or other institutions.

_____

SOH JIEN MIN

Date 28 MARCH 2014

LIST OF FIGURES

LIST OF ABBREVIATIONS/NOTATION/GLOSSARY OF TERMS

Content Management System (CMS)

HyperText Markup Language (HTML)

Extensible Markup Language (XML)

Web Ontology Language (OWL)

Resource Description Framework (RDF)

Resource Description Framework in Attributes (RDFa)

Uniform Resource Identifier (URI)

Web Services Description Language (WSDL)

PHP: Hypertext Preprocessor (PHP)

User Defined XML (UDX)

Application Programming Interface (API)

World Wide Web Consortium (W3C)

What You See Is What You Get (WYSIWYG)

Tiny Moxiecode Content Editor (TinyMCE)

Cascading Style Sheet (CSS)

CHAPTER 1

1.0 INTRODUCTION

With the growing of numbers of websites in the World Wide Web, the issue is no longer locating content and information; rather it is on how to search for them effectively. Search engines have been harvesting and indexing information on all available Web sites for the purpose of locating them through keyword search. However, searching on the basis of simply matching strings is no longer an appropriate approach when accurate, relevant and intuitive results are required. To address this critical drawback, the process of Semantic Search was introduced.[1]

Unlike traditional string match search, where it simply scan the whole page to find a matching string, and output the result base on ranking algorithm, semantic search focus on the page where usually the editor define more meaningful information about the page following different methodologies.[2] One of the few ways to achieve semantic search is known as Resource Description Framework or RDF. The concept of RDF is having 3 components, subject, predicate and object, store in a XML Metadata format and use to store the semantic information, RDF is store alongside with the content that is visible by the website user, providing the key information of the same content in a different format to the search engine.[3]

Several search engines have incorporated Semantic Search as one of the features, but it's not useful until websites begin to have semantic information. Many website still store content in conventional format, whether it's in a static HTML format, or a dynamic content in a database. Even when the content is

stored in database and being defined as dynamic content, when it's being retrieved and displayed in front of the browser they're still nothing more than just a HTML content page. The search engine is doing its job to search through the contents, but the quality relies more on the matter of how the contents are stored and how the content is divided into section that the search engine should look for.

While there are still some legacy websites that update contents purely using a static HTML file, the vast majority of modern websites are managed by some form of Content Management System (CMS). Hence, this project will focus on this type of websites. Content Management System or CMS is a document management that generally known to use to manage web contents, documents for a particular website. In other words, CMS is the interface of how the content the editor wrote and how it's being stored. Most of the contents that is visible in the website will be entered through the system. Currently a lot of work needs to be done by the CMS template developer to insert RDF, and most CMS systems lack of a proper user friendly interface that permits easy insertion. As a result, the template developer will need to expend significant time to accomplish this manually for each website. The primary objective of this project is to develop a CMS that is tailored to work with RDF, thus simplifying interaction of the template developer and also the content editor with it.

We have focused on creating our own CMS in order to address various issues identified in real world cases related to the use of RDF such as allowing content editor to create and save RDF from the user interface and not manually hardcoding them into the HTML, allowing content developer to have a more uniform way to deploy template with RDF. Our custom CMS with RDF-ready features can subsequently be extended to integrate with other CMS's that share a similar core engine; this includes more well-known CMSs such as WordPress.

CHAPTER 2

2.0 LITRITURE REVIEW

2.1 RDF

RDF is a baseline of providing description between information between applications. It contains 3 key components, resources, properties and statements. It's structured and formatted in to XML format. As it is the most portable format to transfer information through different platforms. RDF is basically a framework of how the semantic data should be stored.[4]

Subject (Resource)	http://www.w3.org/Home/Lassila
Predicate (Property)	Creator
Object (literal)	"Ora Lassila"

Figure 1 Example of a RDF parts

Figure 2 Example diagram for Figure 1

Refer to figure 2 with the description "Ora Lassila is the creator of the resource http://www.w3.org/Home/Lassila", a RDF XML can be constructed such as figure 3

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://w3.org/TR/1999/PR-rdf-syntax-19990105#"
xmlns:s="http://description.org/schema/">
<rdf:Description about="http://www.w3.org/Home/Lassila">
<s:Creator>Ora Lassila</s:Creator>
</rdf:Description>
</rdf:RDF>
```

Figure 3 RDF XML Document

2.2 RDF merge with HTML

CMS have been used to manage content and HTML for websites, RDF however is something new to the CMS world. There are already standards of how RDF should be in its own form, but we still need to find out how RDF work together with HTML, as the whole idea is that the same page of content can be understand by both human and machine.[5][6]

2.2.1 Embed RDF into HTML

This approach as it name suggest, simply embed the XML of the RDF straight into the HTML page. Since HTML is basically XML standardised, this would not break the structure of the HTML for most browser. For some older browser however it might be an issue. One example of how this would be presented is a shown in figure 4. The example shown that the RDF structure is put inside the head element of the HTML, it can also be put into the body element of the HTML page as it should not be rendered by the browser as it's not recognize as any visual element. [6]

```
<head>
<title>Some Page</title>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-
ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/">
<rdf:Description rdf:about="http://www.w3.org/" dc:title="W3C
Homepage"/>
</rdf:RDF>
</head>
```

Figure 4 Example of RDF XML in HTML page

This method is simple and straightforward, however, according to [6] it does not help with the validation of the RDF standard. From the HTML point of view, it may or may not break the page standard as newer browser are design to cope with unknown tags, but for some older browser it may break the page. From RDF point of view this is completely invalidate format as it contains a bunch of non-related element for the RDF in the page.

2.2.2 RDFa

Another approach of having RDF in HTML then came later in 2004 and is recommended in 2012 from W3C, is called RDF in attributes. Unlike Embed RDF, it's not putting element into the XML structure of the HTML page. But embed as attributes into the standard HTML elements.

```
<div xmlns:dc="http://purl.org/dc/elements/1.1/"
  about="http://www.example.com/books/wikinomics">
  <span property="dc:title">Wikinomics</span>
  <span property="dc:creator">Don Tapscott</span>
  <span property="dc:date">2006-10-01</span>
</div>
```

Figure 5 Example of RDFa in groups of HTML elements

Looking at figure 5, the RDF information is embedded as attribute of the div element and also for the span element, describing the information of the inner html of the element side by side. One of the main advantage of this approach is that it allows a more close related of where the RDF data with the actual HTML data visually. And it allows multiple RDF information in the same HTML page as it would in having multiple HTML information in the same page.

```
<body vocab="http://purl.org/dc/terms/">
   ...
   <div resource="/alice/posts/trouble_with_bob">
      <h2 property="title">The trouble with Bob</h2>
      <p>Date: <span property="created">2011-09-
10</span></p>
      <h3 property="creator">Alice</h3>
      ...
   </div>
   ...
   <div resource="/alice/posts/jos_barbecue">
      <h2 property="title">Jo's Barbecue</h2>
      <p>Date: <span property="created">2011-09-
14</span></p>
      <h3 property="creator">Eve</h3>
      ...
   </div>
   ...
</body>
```

Figure 6 Example of multiple RDF alongside with multiple HTML

The disadvantage as we see is that it requires some special editor to help with the data population, some WYSIWYG editor such as RDFa Content Editor as oppose to standard editor such as TinyMCE or CKEditor. This means that it requires a specific structuring on the HTML element for the RDF to make sense, thus in some way limiting how the HTML element can be structure which result in limiting how the page can be design. It also requires some knowledge from the content editor to use the editor.[7]

2.2.3 Using Link HTML element

This approach unlike the other, instead of having the RDF in the HTML page, it's trying to have the RDF in its own and simply link back to the HTML page. Utilizing the Link element of HTML standard, much like loading external JavaScript file or external CSS files.[6]

```
<head>
<title>My Document</title>
<link rel="meta" type="application/rdf+xml" href="meta.rdf"/>
</head>
```

Figure 7 Example of meta.rdf link to the HTML page.

This approach avoid the issue with having unknown element in the HTML page, as it's in fact using the known element Link. As well as maintaining the RDF XML to be pure and can be validate. It also inherits the advantage of having external JavaScript file and CSS, where it's only load when needed. This means for ordinary browser it might never be loaded at all, while for other such search engine it might be the only thing to be loaded and thus promote faster and smaller payload.

2.3 Real World Use

2.3.1 Best Buy

Aware of the need of increasing the traffic and visibility, Best Buy US wanted a way to better describe the relationship between products that they sell as well as store information.

The approach that Best Buy uses is RDFa, embedding RDF information into the HTML element of the page. The vocabulary Best Buy uses is Good Relations, a brief look into Good Relations websites reveal that many other also uses Good Relations as the vocabulary reference such as Google and Yahoo!. Good Relations focuses on developing vocabulary that allows describing relationship between web resources, products, prices, terms and conditions.[8][9]

Even though this implementation may not visually benefits for user on the surface, Jay Myers who is the development engineer of Best buy believes that it will enhance the search experience and richer relationship of the product pool and in terms increase the visibility of products.[10]

2.3.2 Drupal

Drupal is an open source CMS that is written in PHP. It is currently used by 1.9% of all website according to Web Technology Survey.[11] Michael Anello from Drupal, highlighted that even thou there isn't high volume of request for Drupal to venture into the RDF from their client, there are a number of good reason he wanted to implement them in the latest version of Drupal. This include the better search ranking when the content is populate with a meaningful context, as well as being able to link with other content that have the same context, bringing more visibility to the clients websites.[12][13]

2.3.3 Yahoo!

Yahoo! SearchMonkey is Yahoo!'s approach to Semantic Search via structural data such as RDF. It will crawl website for their RDFa and index along with the unstructured content, which is the page itself, to enhance the search result.[14] This service however was shutdown in 2010 when as part of the deal between Microsoft and Yahoo!.[15]

Later on Yahoo! release a demo search engine, specifically RDF search engine called Glimmer. It allows searches of RDF base on 750 million triplets that uses schema.org, a collaboration site between Bing, Google and Yahoo! that provides collections of schemas and vocabularies shared and used between other search engines such as Google and Bing.[16] [17]

2.3.4 Google Rich Snippets!

Google Rich Snippets was release to display structural data in their search result, showing user a summary of the page, enriching the search result. It works by looking at the RDFa inside the page. Following a set of standard annotations, it is able to define what kind of snippet can be display in the search result such as reviews, music, calendar events.[18]

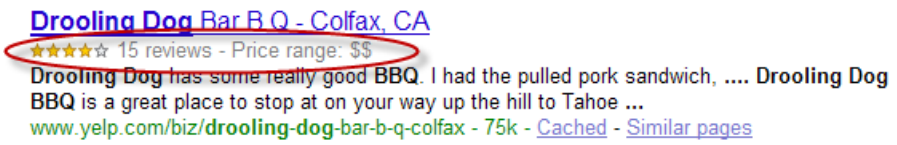

Figure 8 Example of Rich Snippets

2.4 RAP

RAP is a RDF API written in PHP, specifically for PHP developers. It provides two API called Model API and ResModel API and supports adding, deleting and replacing statements inside a Model. It was introduce by Freie Universität Berlin in 2002 as an opensource project. The latest version we found is V0.9.6. The RAP have documentation and we will be using several API such as DBStore, getModel and so on to store our RDF.[19]

CHAPTER 3

3.0 DESIGN

In this research we will be using a CMS called EVO. The CMS is completely services based, the front end of the CMS is build using ADOBE Flex technology and the backend service we're using PHP and MySQL for the database. For the RDF part, we will be using RAP to merge with the CMS, storing the RDF data and generating RDF for frontend using RAP. RAP is also base on PHP language so it should be much more compatible and at the same time open source.

The CMS is intent to manage many different kind of content, so the first thing we need to make sure is to not have any kind of hardcode in resource type. To do this, the management of the semantic data have to be dynamic and configurable; it basically breaks down into two sections, administrator who creates any kind of resource type and editor who chooses them as fields to populate when editing content. In figure 9 show the two main user involve, the Administrator and Content Editor.

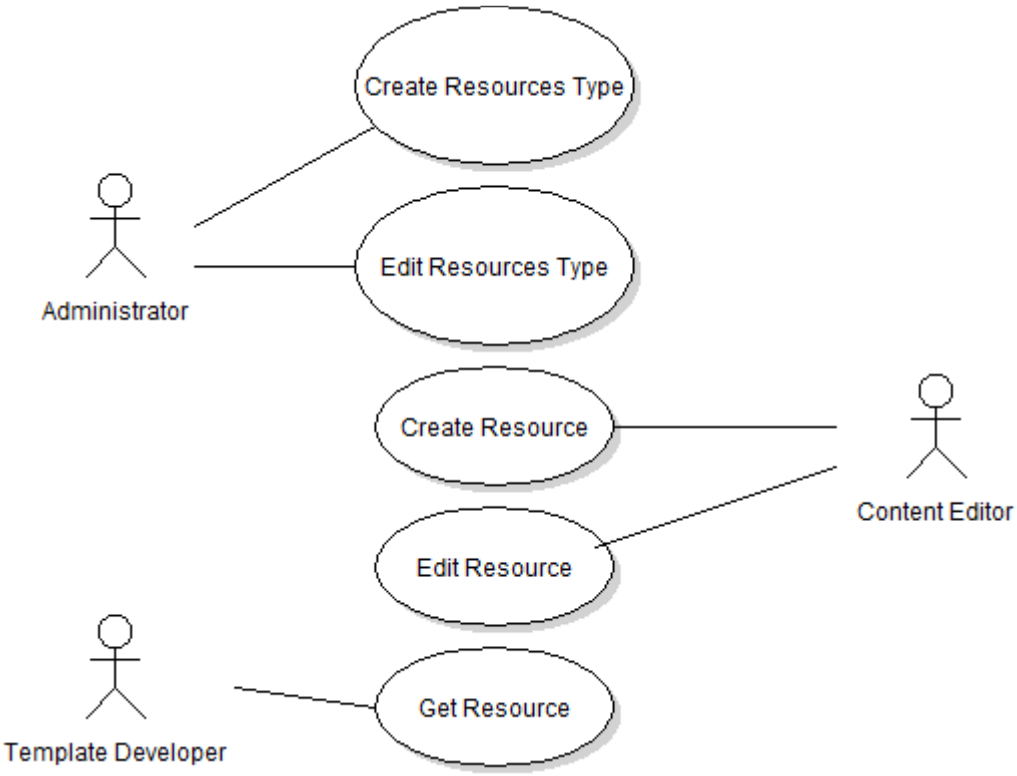

Figure 9 Use Case Diagram

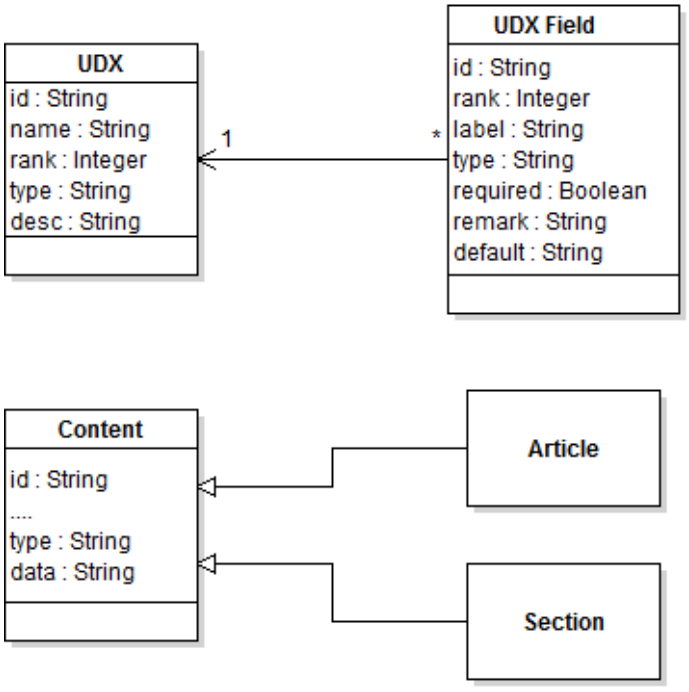

Figure 10 Class Diagram

3.1 Administration

The original structure of the CMS is using a folder/file structure, where a section is a container that contains multiple articles, every section can also have child section, so it forms a tree like structure, this allow very flexible web sitemap design. For example if the website needs to store information of books, we can create a root section call books, and then have children section that separate by category, then in each category section there can be articles that store the book information, such as description and images. This is where we will be tapping in the resource, by associating every semantic resource on the actual article itself, it allows very easy maintenance.

The CMS will have an interface where user with administration responsible will create multiple different type of resource, every resource will have multiple information, for ease of use there can be different type of user interface for example text box, check box, dropdown and more. This allows more user friendly configuration for editor to fill in and also more control and consistency when creating resource. Continue on the example of a books, administrator will be creating a resource type call book, and add several information field for the book type, such as title and author, they can also be URIs. In the CMS we are calling this mode UDX, it stand for User Defined XML.

Figure 11 and 12 describe the activity when administrator create or edit a UDX type. The UDX Admin Mode is only available to the administrator and should not be available to the content editor.

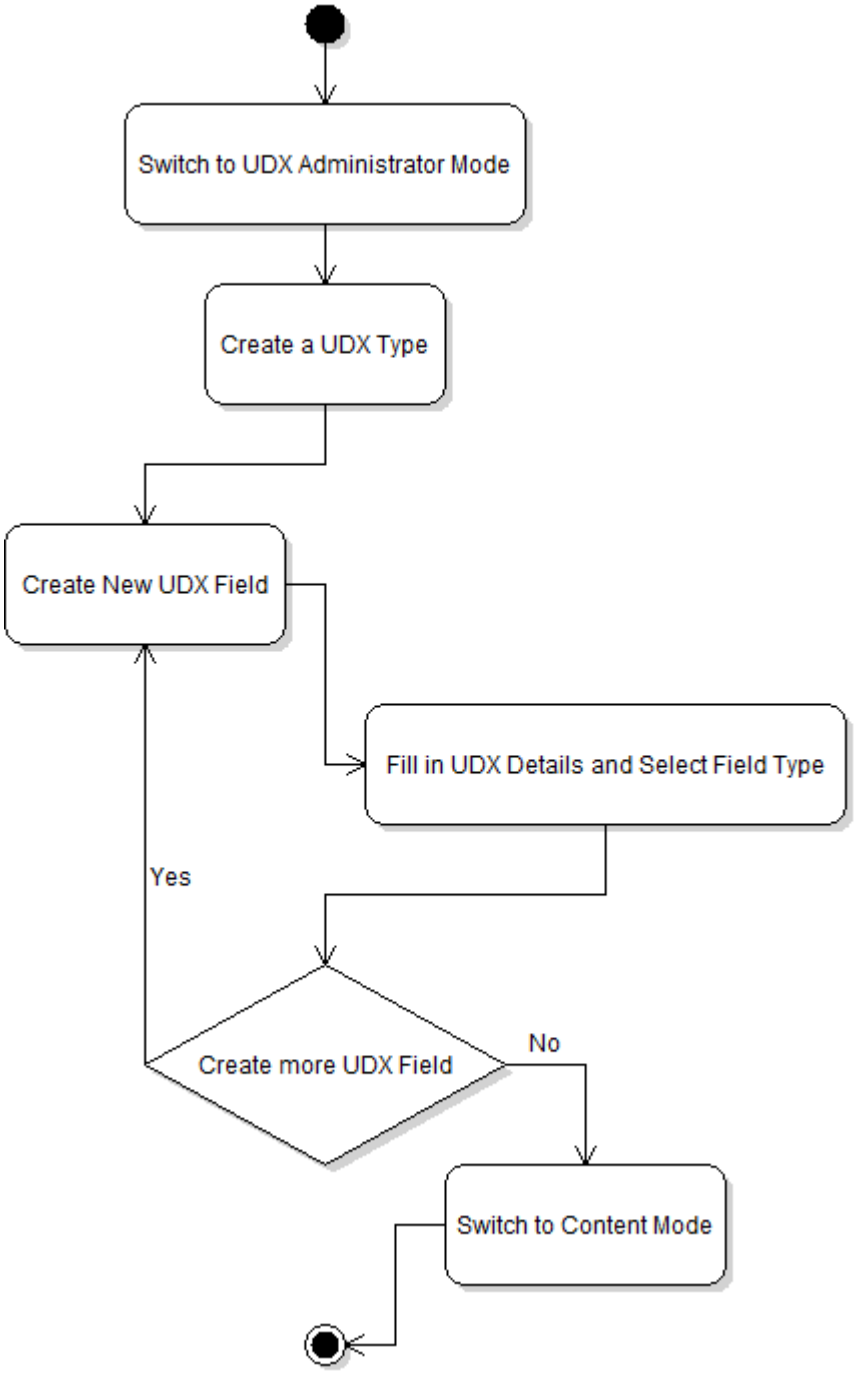

Figure 11 Activity Diagram for Create Resources Type

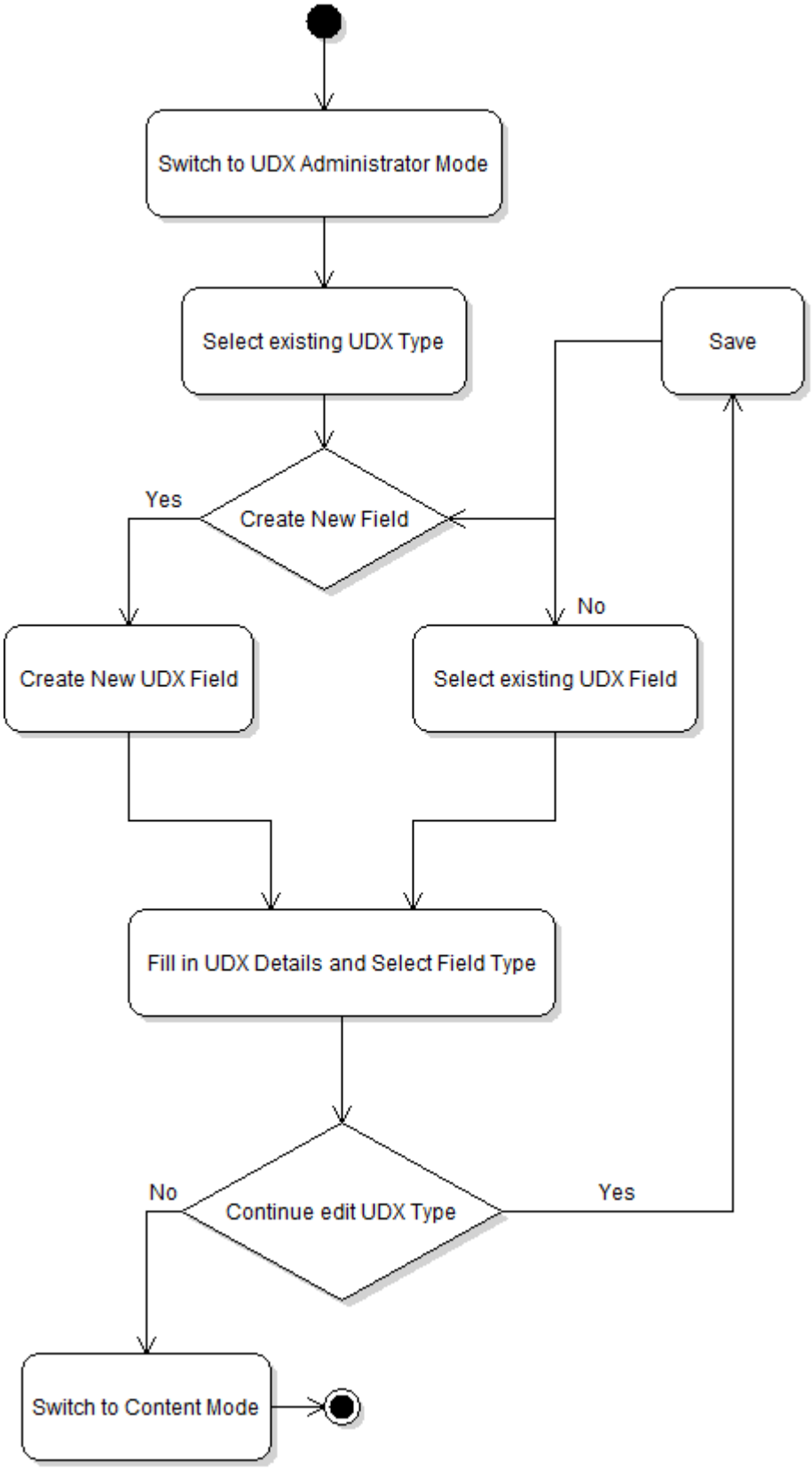

Figure 12 Activity Diagram for Edit Resource Type

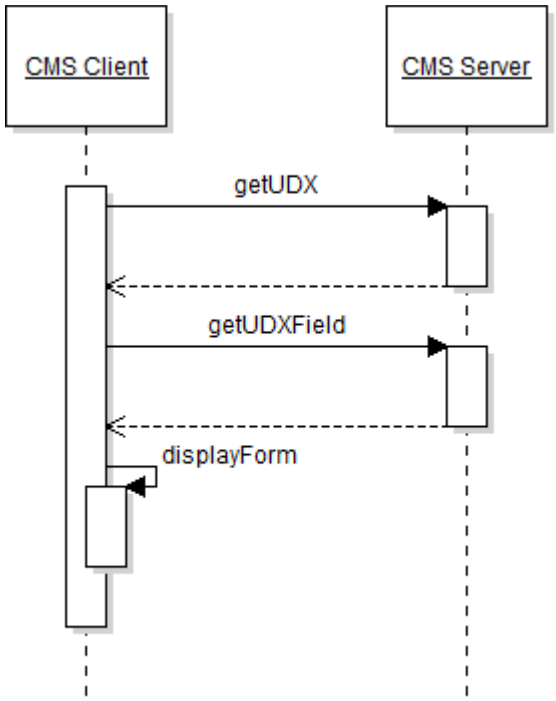

Figure 13 Sequence when Administrator read UDX

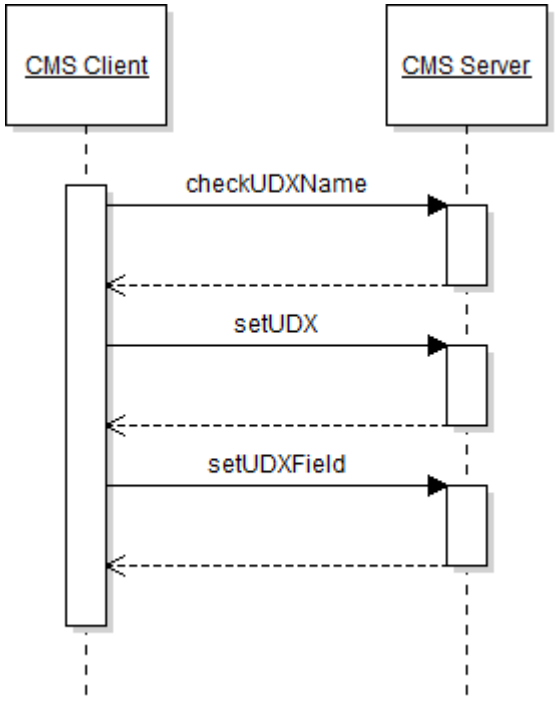

Figure 14 Sequence when Administrator save UDX

3.2 Content Editor

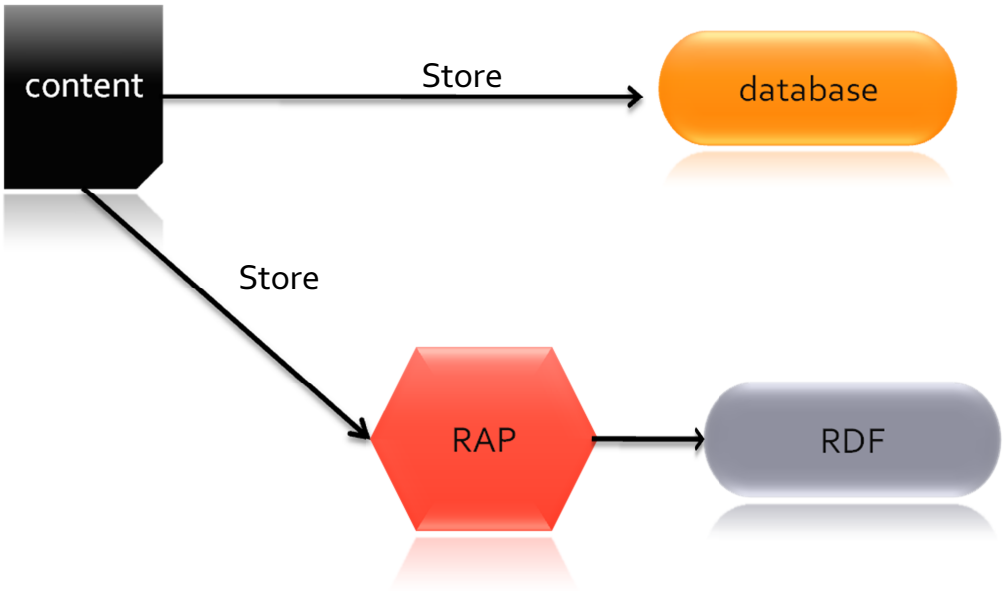

Figure 15 Simple Concept of Content Storage

When editor starts to create or update content, they will navigate into the proper article or section, from there they will have to choose what resource type is this article about, if it's for a book then they will be selecting a book, which are created by the administrator. By selecting the type, the associated field will load and editor can fill in the values for those fields or URIs. If a dropdown field or checkbox is used then user will select appropriate values that are predefined. A shown in figure 15, content will be stored on both the database and in the RAP database.

When editor select a section or an article for edit, it will retrieve content from the server and process the XML. At the same time it will retrieve the UDX information that the administrator have save and generate the UDX fields, as show in figure 17. When a section or an article is save, the XML will first be

generated and submit to the server, the server will automatic update the RAP within itself at the same time generate the RDF file. Refer to figure 18.

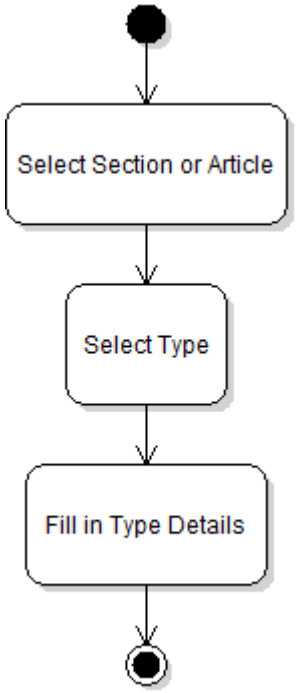

Figure 16 Activity Diagram for Create and Edit Resource

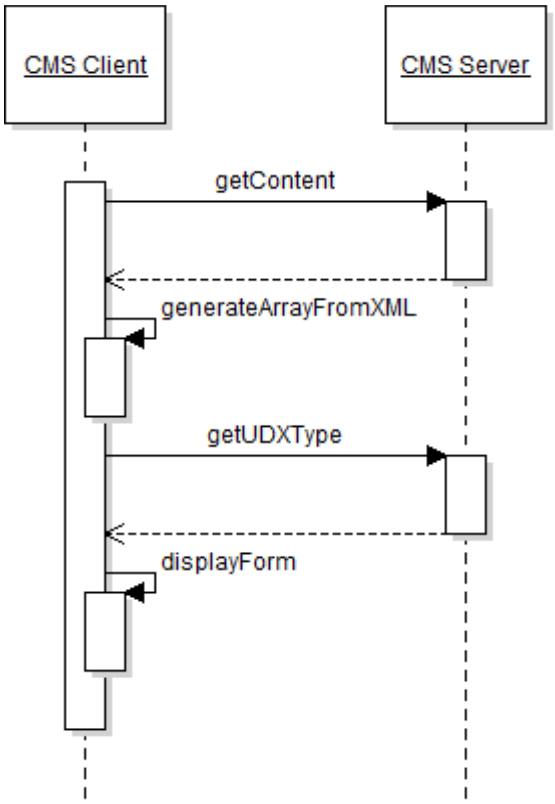

Figure 17 Content Editor Read Content

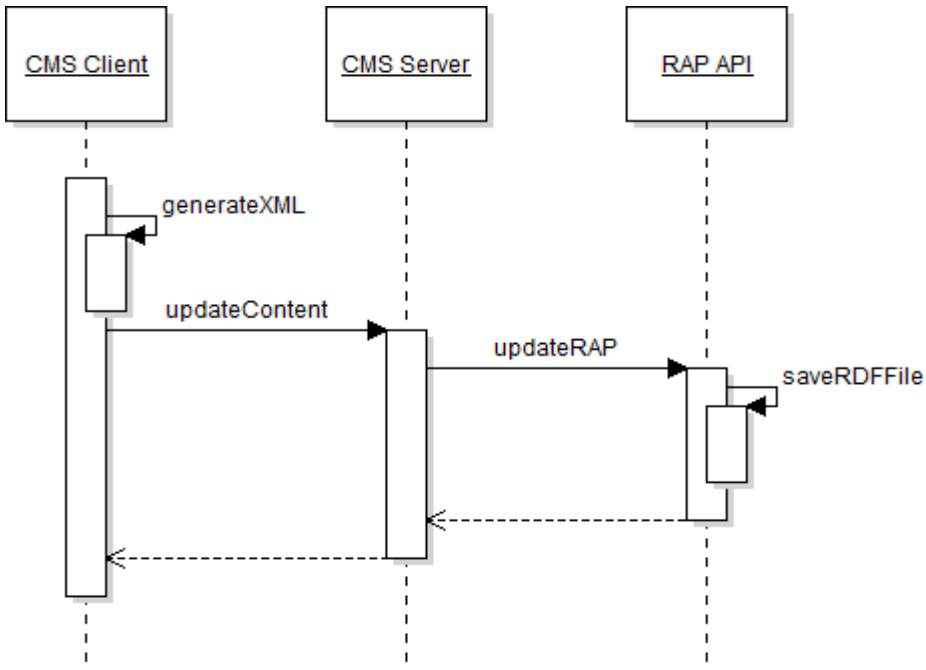

Figure 18 Content Editor Update Content correspond to Figure 15

3.3 Template

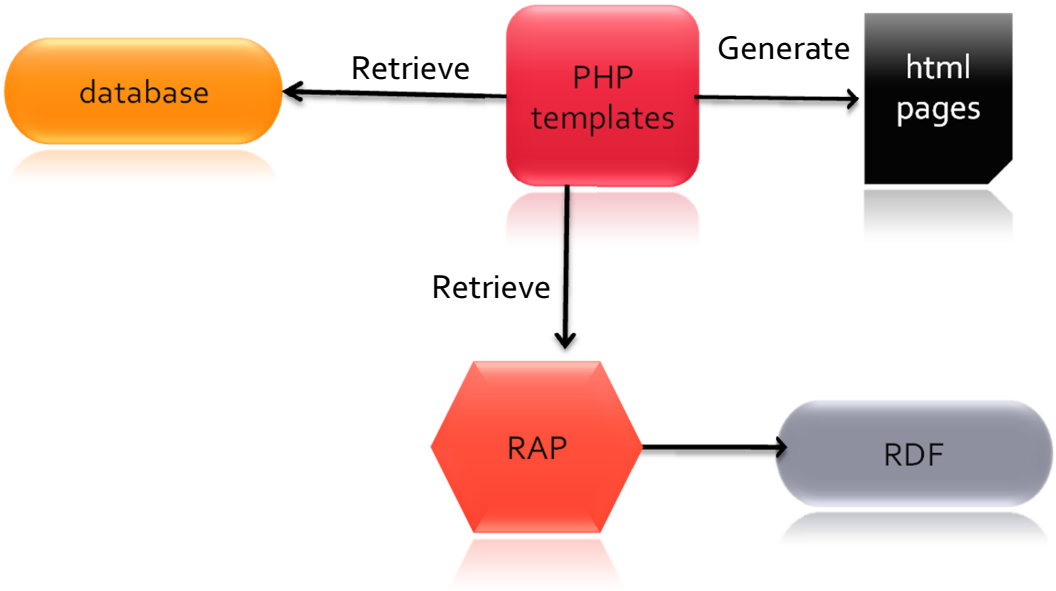

Figure 19 Content Retrieval

During the template development of a website, the developer will be calling a function to retrieve the RDF file and place in the HTML page. We will be using the link element as describe in chapter 2.2.3. Since the file are created earlier whenever there is an update to the content, as in figure 20, there is no need for the RDF to create real time, reducing the processing power require otherwise.

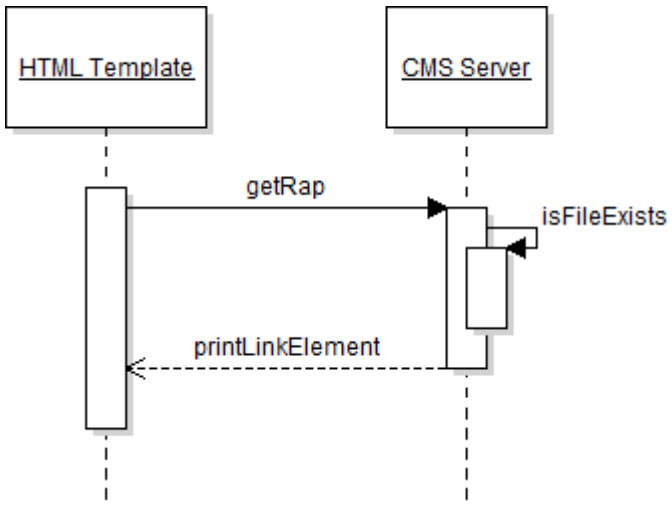

Figure 20 HTML RDF Link Element Correspond to Figure 19

CHAPTER 4

4.0 DEVELOPMENT

Due to the requirement of able to fulfil any kind of generic information, a dynamic field and value pair is essential, which means the traditional predefined table field and column would not work. So instead of storing data into table directly, we are using XML to store the dynamic structure and data into database.

4.1 UDX Template

While the actual data is stored using XML, the XML template are still store info database as refer to Class Diagram figure 10. So we created 2 tables to do so, tbl140_xml and tbl141_struct. tbl140_xml is responsible to store the resource, for example a book. Each resource will have different fields, which is store in tbl141_struct, in this table each field can be store with form type such as textbox, checkbox and combo box. This make up for the XML template that require for populating the content.

Column	Type	Null	Default	Comments
t140_xmlid	varchar(36)	No		
t140_name	varchar(255)	Yes	*NULL*	
t140_rank	smallint(5)	Yes	*NULL*	
t140_type	varchar(255)	Yes	*NULL*	
t140_desc	varchar(255)	Yes	*NULL*	
t140_createdby	varchar(36)	Yes	*NULL*	
t140_dt_created	datetime	Yes	*NULL*	
t140_deleted	tinyint(1)	Yes	0	
t140_deletedby	varchar(36)	Yes	*NULL*	
t140_dt_deleted	datetime	Yes	*NULL*	

Figure 21 tbl140_xml

Column	Type	Null	Default	Comments
t141_structid	varchar(36)	No		
t141_xmlid	varchar(36)	Yes	*NULL*	
t141_rank	smallint(5)	Yes	*NULL*	
t141_label	varchar(255)	Yes	*NULL*	
t141_type	varchar(255)	Yes	*NULL*	
t141_required	tinyint(1)	Yes	0	
t141_remark	longtext	Yes	*NULL*	
t141_default	longtext	Yes	*NULL*	
t141_createdby	varchar(36)	Yes	*NULL*	
t141_dt_created	datetime	Yes	*NULL*	
t141_deleted	tinyint(1)	Yes	0	
t141_deletedby	varchar(36)	Yes	*NULL*	
t141_dt_deleted	datetime	Yes	*NULL*	

Figure 22 tbl141_struct

We've also created a module in Adobe Flex for easy manage of these XML template in which we call it UDX. This allows the creation of new resource and fields, as well as configuring the suitable form for easy input. The process of managing UDX is as figure 11 and figure 12. Administrator first switch to UDX mode and then depending on situation create or edit existing UDX, adding or editing UDX fields.

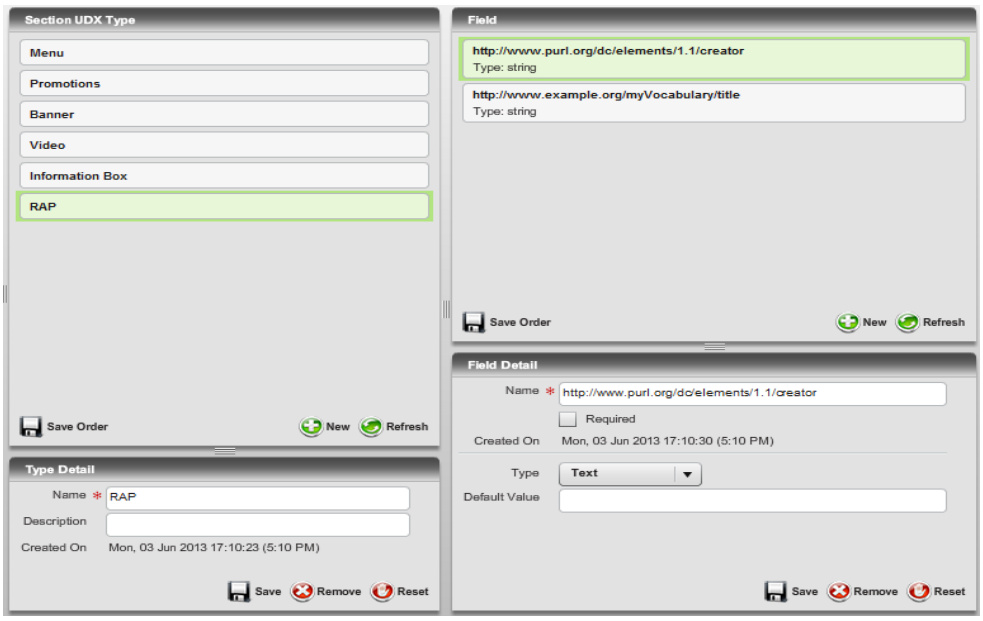

Figure 23 UDX Mode in EVO

Types of form field include text, number, browse, date and time, check box, combo box and hidden, which depending on cases can also have default or predefined value. A note here is browse is for browsing of the CMS internal file system for files, images or links.

Along to that we've also created the API to serve the UDX editor as well as XML data. This API includes updating the UDX and its fields. The processes are based on figure 13 and figure 14. Refer to figure 13, the code name of the UDX needs to be unique hence the checking before the update is required. Hence we created the checking as a separate API.

4.2 UDX Data

Referring to figure 17, to store data using the UDX, another module call _form_udx.mxml need to be created in Adobe Flex. The standard module will take the UDX resource id, retrieve the fields, and generate the interface for user to input data. It will be a standard component that can be use anywhere in CMS as long as the id is provided. In our case we are going to use them in Sections and Articles. So we insert the module into these two forms. At the same time created the require fields for the two table. T40_xml_type_id and t40_xml is added to tbl40_section and t50_xml_type_id is added to tbl50_article and t51_xml is added to tbl51_aversion. Generally xml_type holds the id to the UDX resource, while xml store the actual data in xml format.

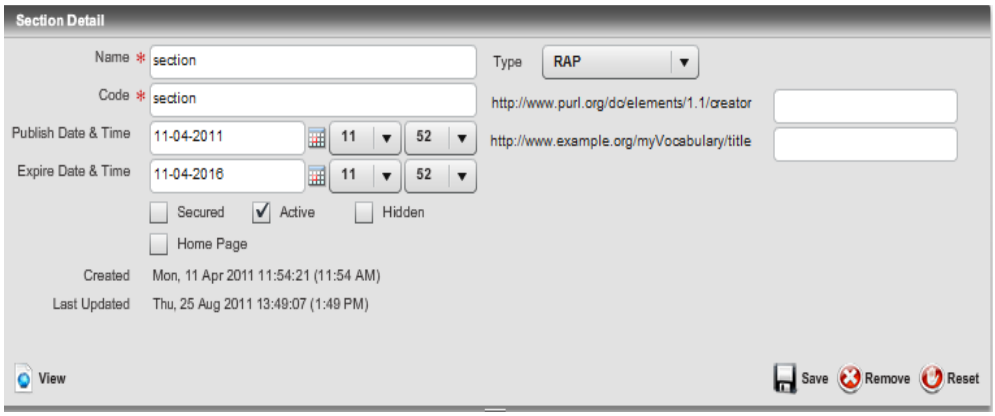

Figure 24 Example of section detail with RAP type

When a section or an article is save, the UDX data will be generated and save, just as describe in figure 18. The XML data structure would have the XML format like figure 25, and each field is repeated, with t141_structid being the field id of the xml template tbl141_stuct.

```
<field>
    <t141_structid>7465f031-5b68-11e0-afd1-5c3a10a0f3df</t141_structid>
    <data>some data</data>
</field>
```

Figure 25 Example of xml data structure

t40_xml_typeid	t40_xml
c900b0a5-eb2b-102b-8453-bc9cda71a5e8	<field><t141_structid>c8a5b12b-59f0-11e0-8263-a903114d4fa4</t141_structid><data><![CDATA[FAQs]]></data></f
c900b0a5-eb2b-102b-8453-bc9cda71a5e8	<field><t141_structid>c8a5b12b-59f0-11e0-8263-a903114d4fa4</t141_structid><data><![CDATA[Hong Kong]]></dat
c900b0a5-eb2b-102b-8453-bc9cda71a5e8	<field><t141_structid>c8a5b12b-59f0-11e0-8263-a903114d4fa4</t141_structid><data><![CDATA[Neptune Login]]><
c900b0a5-eb2b-102b-8453-bc9cda71a5e8	<field><t141_structid>c8a5b12b-59f0-11e0-8263-a903114d4fa4</t141_structid><data><![CDATA[Hong Kong Side]]>
6cec0645-5b68-11e0-afd1-5c3a10a0f3df	<field><t141_structid>7465f031-5b68-11e0-afd1-5c3a10a0f3df</t141_structid><data><![CDATA[Transfer Service]]><
c900b0a5-eb2b-102b-8453-bc9cda71a5e8	<field><t141_structid>c8a5b12b-59f0-11e0-8263-a903114d4fa4</t141_structid><data><![CDATA[Taiwan]]></data><
6cec0645-5b68-11e0-afd1-5c3a10a0f3df	<field><t141_structid>7465f031-5b68-11e0-afd1-5c3a10a0f3df</t141_structid><data><![CDATA[Hotel Promotions]]><

Figure 26 Example of udx being stored

4.2 RAP

To store RDF, we're using RAP as the library. After including the RAP required file, we created a new internal API rap.php. The API takes the data object directly from section and article and proceeds to create the RDF for them. Existence of RAP database object are also check before continue.

We then proceed to call this API from the section.php API and article.php API to facilitate and start storing into RAP. We utilize the function call DBStore to create a database object for RAP, then by using modelExists function to check for an existing model we determine to call getModel or getNewModel to retrieve or create the model. And having the model object we can then add statement into the model. After adding the statement, we call the saveAs API to output a rdf file. Doing so will ensure that every time there is an update, the file will be updated as well. Process was described in figure 15 and figure 18.

As figure 19 and figure 20, to proceed with allowing template developer to output the RDF file, we've created a function call get_rap and it will check when the file is exists, it will output the link element and link to that file in the HTML output.

CHAPTER 5

5.0 CONCLUSION AND FUTURE WORK

The immediate goal of this research is for our CMS to be able to take advantage of RDF and make websites it manages ready for the Semantic Search. We've successfully made the CMS able to accept dynamic range of RDF created by the template developer, as well as easy populate by the content editor.

As mention in the introduction, because our use of CMS that build upon PHP and MySQL, we will look into using the things that we learn from this research and implementing them into other CMS, such as WordPress and MODX. A plugin can potentially be developed for WordPress as well as MODX due to their mature plugin integration support. Also due to the core concept of WordPress used of metadata and MODX used of generic table index for dynamic fields, the structure have great potential and possibility to have a similar integration with RDF using RAP.

REFERENCE

[1] Nigel Shadbolt, Wendy Hall, and Tim Berners-Lee, “The Semantic Web Revisited,” IEEE Intelligent Systems, pp. 96-101, Jun. 2006.

[2] Hamid Haidarian Shahri, “Semantic Search in Linked Data: Opportunities and Challenges” AAAI, 2010.

[3] Jeremy J. Carroll and Patrick Stickler, “RDF triples in XML,” WWW Alt.'04, pp. 412-413, 2004.

[4] Ora Lassila and Ralph R. Swick, “Resource Description Framework (RDF) Model and Syntax Specification,” World Wide Web Consortium, 1998.

[5] Tim Berners-Lee, “RDF in HTML,” www.w3.org, Apr. 17, 2002. [Online]. Available: http://www.w3.org/2002/04/htmlrdf. [Accessed: Jan. 11, 2014].

[6] Sean B. Palmer, “RDF in HTML: Approaches,” www.w3.org, June. 2, 2002. [Online]. Available: http://infomesh.net/2002/rdfinhtml/. [Accessed: Jan. 11, 2014].

[7] Ivan Herman, Ben Adida, Manu Sporny and Mark Birbeck, “RDFa 1.1 Primer - Second Edition,” World Wide Web Consortium, Aug. 23, 2013.

[8] Martin Hepp, “GoodRelations: An Ontology for Describing Products and Services Offers on the Web” Knowledge Engineering: Practice and Patterns, pp 329-346, 2008.

[9] “Prominent Users of GoodRelations,” wiki.goodrelations-vocabulary.org, Apr. 9, 2013. [Online]. Available: http://wiki.goodrelations-vocabulary.org/References. [Accessed: Jan. 10, 2014].

[10] Richard MacManus, “How Best Buy is Using The Semantic Web,” readwrite.com, Jun. 30, 2010. [Online]. Available: http://readwrite.com/2010/06/30/how_best_buy_is_using_the_semantic_web. [Accessed: Dec. 15, 2013].

[11] “Usage of content management systems for websites,” w3techs.com, Mar. 24, 2014. [Online]. Available: http://w3techs.com/technologies/overview/content_management/all. [Accessed: Mar. 24, 2014].

[12] Michael Anello, “RDF in Drupal: What is it and Why Should We Care?” drupaleasy.com, Jun. 15, 2009. [Online]. Available: http://drupaleasy.com/blogs/ultimike/2009/06/rdf-drupal-what-it-why-should-we-care. [Accessed: Feb. 02, 2014].

[13] Stéphane Corlosquet, Renaud Delbru, Tim Clark, Axel Polleres and Stefan Decker, “Produce and Consume Linked Data with Drupal!” The Semantic Web - ISWC 2009, pp 763-778, 2009.

[14] Tim Finin, “Yahoo! adds RDF support to SearchMonkey and BOSS” ebiquity.umbc.edu, Feb. 12, 2009. [Online]. Available: http://ebiquity.umbc.edu/blogger/2009/02/12/yahoo-adds-rdf-support-to-searchmonkey-and-boss/. [Accessed: Feb. 02, 2014].

[15] Tom Krazit, “Yahoo starts Bing transition, kills Search Monkey” cnet.com, Aug. 17, 2010. [Online]. Available: http://www.cnet.com/news/yahoo-starts-bing-transition-kills-search-monkey/ [Accessed: Feb. 02, 2014].

[16] Roi Blanco, Peter Mika, and Sebastiano Vigna, “Effective and Efficient Entity Search in RDF data,” Yahoo! Research, 2011.

[17] Aaron Bradley, “RDF Search Engine Glimmer Released by Yahoo” seoskeptic.com, Jun. 20, 2013. [Online]. Available: http://www.seoskeptic.com/rdf-search-engine-glimmer-released-by-yahoo/ [Accessed: Feb. 02, 2014].

[18] Kavi Goel, Ramanathan V. Guha, and Othar Hansson, “Introducing Rich Snippets” googlewebmastercentral.blogspot.com, May. 12, 2009. [Online]. Available: http://googlewebmastercentral.blogspot.com/2009/05/introducing-rich-snippets.html [Accessed: Feb. 02, 2014].

[19] Daniel Westphal and Chris Bizer, “Introduction to RAP” uni-mannheim.de, Oct. 2004. [Online]. Available: http://wifo5-03.informatik.uni-mannheim.de/bizer/rdfapi/tutorial/introductionToRAP.htm [Accessed: Dec. 01, 2013].

```php
<?PHP
define("RDFAPI_INCLUDE_DIR", dirname(__FILE__)."/../../../rap/rdfapi-php/api/");
include_once "system.php";
include(RDFAPI_INCLUDE_DIR . "RdfAPI.php");

function rap(){
    function get($data){
    }
    function add($data){
        if(array_key_exists('t40_sectionid', $data)){
            $arr = get_udx($data['t40_xml_typeid'], $data['t40_xml']);
            $modelURI = 'section';
            $baseURI = 'http://www.example.org/someDocument.html';
            $subject = $data['t40_sectionid'];
        }
        else if(array_key_exists('t50_articleid', $data)){
            $arr = get_udx($data['t50_xml_typeid'], $data['t51_xml']);
            $modelURI = 'article';
            $baseURI = 'http://www.example.org/someDocument.html';
            $subject = 'http://www.example.org/someDocument.html';
        }

        if(!empty($arr)){
            $rdf_database = new DbStore('MySQL', 'localhost', 'uniasia_db', 'root',
             'p@$$w0rd' );
            if ($rdf_database->modelExists($modelURI)){
                $dbModel = $rdf_database->getModel($modelURI);
            }
            else{
                $dbModel = $rdf_database->getNewModel($modelURI, $baseURI);
            }
            $lang = $arr['Language'];
            foreach($arr AS $key => $value){
                if($key != 'Language'){
                    $statement = new Statement(new Resource($subject),
                                               new Resource(trim($key)),
                                               new Literal(trim($value), $lang));
                    $dbModel->add($statement);
                    $dbModel->saveAs(dirname(__FILE__)."/../../../rdf/".$subject.
                    ".rdf");
                }
            }
            $dbModel->close();
        }
    }
    function update($data){
        add($data);
    }
    function remove($data){
    }
}
?>

```

```php
<?PHP
include_once "_bin/include/system.php";

function get_rap(){
    $section = get_section($_REQUEST['sc'],"array","",0);
    $section = $section['data'];
    if(!empty($section)){
        $section = $section[0];
        if(file_exists(dirname(__FILE__)."/../../../rdf/".$section['id'].".rdf")){
            echo '<link rel="meta" type="application/rdf+xml" href="../rdf/'.
            $section['id'].'.rdf"/>';
        }
    }
    $article = get_article("",$_REQUEST['ar'],"",true);
    $article = $article['data'];
    if(!empty($article)){
        $article = $article[0];
        if(file_exists(dirname(__FILE__)."/../../../rdf/".$article['id'].".rdf")){
            echo '<link rel="meta" type="application/rdf+xml" href="../rdf/'.
            $article['id'].'.rdf"/>';
        }
    }
}

?>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ns:_ui_hdividedbox creationComplete="init()" xmlns:mx=
"http://www.adobe.com/2006/mxml" xmlns:ns="_com.*">
    <mx:Script>
        <![CDATA[
            import mx.events.DragEvent;
            import mx.events.ListEvent;
            import mx.rpc.events.*;

            private var HTTPService:_sys_httpService;
            public var parentObject:Object;
            public var targetType:String;
            private var selectedId:Number=-1;
            private var selectedId2:Number=-1;
            //[Bindable]private var app:_app = new _app;

            private function init():void{
                ui();
                currentState = targetType;
                if(targetType == "member" || targetType == "user"){
                    getTargetXML(targetType);
                }
                else{
                    getXML();
                }
                //selRole();
            }
            private function ui():void{
            }
            public function chkXML(formParam:Object, resultHandler:Function):void{
                var HTTPparams:Object = new Object;
                HTTPparams = formParam;
                HTTPparams.func="cUDXXML.chk";
                HTTPparams.t140_type = targetType;
                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, resultHandler);
            }
            public function getXML(newId:String=""):void{
                var HTTPparams:Object = new Object;
                HTTPparams.func="cUDXXML.get";
                HTTPparams.t140_type = targetType;

                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, resultHandler);

                function resultHandler(event:ResultEvent):void{
                    if(!_app.resultStatus(event)){return;}

                    var newArr:Array=_app.getContentData(event);
                    var i:Number = -1;
                    for each(var item:Object in newArr){
                        i++;
                        if(item.t140_xmlid==newId){
                            selectedId = i;
                        }
                    }
```

```
54                      xmlList.data=newArr;
55                      xmlList.callLater(defaultListIndex);
56                  }
57              }
58              private function getTargetXML(typeIn:String):void{
59                  var HTTPparams:Object = new Object;
60                  HTTPparams.func="cUDXXML.get";
61                  HTTPparams.t140_type = typeIn;
62                  HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
63
64                  function resultHandler(event:ResultEvent):void{
65                      if(!_app.resultStatus(event)){return;}
66
67                      var newArr:Array=_app.getContentData(event);
68                      xmlList.data=newArr;
69                      selectedId = 0;
70                      xmlList.callLater(defaultListIndex);
71                  }
72              }
73              private function defaultListIndex():void{
74                  xmlList.selectedIndex=selectedId;
75                  xmlList.dispatchEvent(new ListEvent(ListEvent.CHANGE));
76              }
77              private function selectXML(deselect:Boolean = false):void{
78                  selectedId = xmlList.selectedIndex;
79                  if(xmlList.selectedItem && !deselect){
80                      getField();
81                      typeForm.data = xmlList.selectedItem;
82                  }else{
83                      fieldList.data = null;
84                      typeForm.data = null;
85                  }
86              }
87              public function addXML(formParam:Object):void{
88                  var HTTPparams:Object = new Object;
89                  HTTPparams = formParam;
90                  HTTPparams.func="cUDXXML.add";
91                  HTTPparams.t140_type= targetType;
92
93                  HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, xmlResultHandler, null, null, true, true);
94              }
95              public function updateXML(formParam:Object):void{
96                  var HTTPparams:Object = new Object;
97                  HTTPparams = formParam;
98                  HTTPparams.func="cUDXXML.update";
99                  HTTPparams.t140_type= targetType;
100
101                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, xmlResultHandler, null, null, true, true);
102             }
103             public function removeXML(formParam:Object):void{
104                 var HTTPparams:Object = new Object;
105                 HTTPparams = formParam;
106                 HTTPparams.func="cUDXXML.remove";
```

```
107
108                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, xmlResultHandler, null, null, true, true);
109             }
110             private function rankXML():void{
111                 var rank:String = "";
112                 var i:Number = 1;
113
114                 rank += "<root>";
115                 for each(var item:Object in xmlList.dataProvider){
116                     rank += '<item id="' + item.t140_xmlid + '" rank="' + i + '"/>';
117                     i++;
118                 }
119                 rank += "</root>";
120
121                 var HTTPparams:Object = new Object;
122                 HTTPparams.xml_data = rank;
123                 HTTPparams.func="cUDXXML.rank";
124
125                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, xmlResultHandler, null, null, true, true);
126             }
127             private function xmlResultHandler(event:ResultEvent):void{
128                 if(!_app.resultStatus(event)){return;}
129
130                 getXML(_app.resultReturn(event));
131             }
132 /*
133     field start
134 */
135
136             public function chkField(formParam:Object, resultHandler:Function):void{
137                 var HTTPparams:Object = new Object;
138                 HTTPparams = formParam;
139                 HTTPparams.func="cUDXField.chk";
140                 HTTPparams.t141_xmlid = xmlList.selectedItem.t140_xmlid;
141                 //HTTPparams.t141_structid = fieldList.selectedItem.t141_structid;
142
143                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
144             }
145             public function getField(newId:String=""):void{
146                 //selectField(true);
147
148                 var HTTPparams:Object = new Object;
149                 HTTPparams.func="cUDXField.get";
150                 HTTPparams.t141_xmlid = xmlList.selectedItem.t140_xmlid;
151
152                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
153
154                 function resultHandler(event:ResultEvent):void{
155                     if(!_app.resultStatus(event)){return;}
156
157                     var newArr:Array=_app.getContentData(event);
158                     var i:Number = -1;
```

```
159                     for each(var item:Object in newArr){
160                         i++;
161                         if(item.t141_structid==newId){
162                             selectedId2 = i;
163                         }
164                     }
165                     fieldList.data=newArr;
166                     fieldList.callLater(defaultFieldListIndex);
167                 }
168             }
169             private function defaultFieldListIndex():void{
170                 fieldList.selectedIndex=selectedId2;
171                 fieldList.dispatchEvent(new ListEvent(ListEvent.CHANGE));
172             }
173             private function selectField(deselect:Boolean = false):void{
174                 selectedId2 = fieldList.selectedIndex;
175                 if(fieldList.selectedItems.length > 0 && !deselect){
176                     fieldForm.data = fieldList.selectedItem;
177                 }else{
178                     fieldForm.data = null;
179                 }
180             }
181             public function addField(formParam:Object):void{
182                 var HTTPparams:Object = new Object;
183                 HTTPparams = formParam;
184                 HTTPparams.func="cUDXField.add";
185                 HTTPparams.t141_xmlid= xmlList.selectedItem.t140_xmlid;
186 
187                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, fieldResultHandler, null, null, true, true);
188             }
189             public function updateField(formParam:Object):void{
190                 var HTTPparams:Object = new Object;
191                 HTTPparams = formParam;
192                 HTTPparams.func="cUDXField.update";
193 
194                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, fieldResultHandler, null, null, true, true);
195             }
196             public function removeField(formParam:Object):void{
197                 var HTTPparams:Object = new Object;
198                 HTTPparams = formParam;
199                 HTTPparams.func="cUDXField.remove";
200 
201                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, fieldResultHandler, null, null, true, true);
202             }
203             private function rankField():void{
204                 var rank:String = "";
205                 var i:Number = 1;
206 
207                 rank += "<root>";
208                 for each(var item:Object in fieldList.dataProvider){
209                     rank += '<item id="' + item.t141_structid + '" rank="' + i +
                        '"/>';
210                     i++;
```

```
211                 }
212                 rank += "</root>";
213 
214                 var HTTPparams:Object = new Object;
215                 HTTPparams.xml_data = rank;
216                 HTTPparams.func="cUDXField.rank";
217 
218                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, fieldResultHandler, null, null, true, true);
219             }
220             private function fieldResultHandler(event:ResultEvent):void{
221                 if(!_app.resultStatus(event)){return;}
222 
223                 getField(_app.resultReturn(event));
224                 //_app.error(_app.resultReturn(event));
225             }
226 
227         ]]>
228     </mx:Script>
229     <ns:_ui_vdividedbox width="45%" id="xmlContainer">
230         <ns:_ui_panel_container>
231             <ns:_ui_panel title="Type" id="xmlListContainer">
232                 <ns:_ui_list id="xmlList" setState="udx_XML" columnCount="1"
                    dragFormat="xmlList" dragEnabled="true" dropEnabled="true"
                    dragMoveEnabled="true" height="100%" width="100%" parentObject=
                    "{this}" change="selectXML();" dragStart="selectXML(true);"/>
233                 <ns:_ui_btn_bar>
234                     <ns:_ui_btn currentState="saveOrderBtn" click="rankXML();" />
235                     <mx:Spacer width="100%"/>
236                     <ns:_ui_btn id="newXMLBtn" currentState="newBtn" click=
                        "typeForm.newBtn.dispatchEvent(new
                        MouseEvent(MouseEvent.CLICK));"/>
237                     <ns:_ui_btn currentState="refreshBtn" click="init();" />
238                 </ns:_ui_btn_bar>
239             </ns:_ui_panel>
240         </ns:_ui_panel_container>
241         <ns:_ui_panel_container height="140">
242             <ns:_ui_panel title="Type Detail">
243                 <ns:udx_type_form id="typeForm" parentObject="{this}"/>
244             </ns:_ui_panel>
245         </ns:_ui_panel_container>
246     </ns:_ui_vdividedbox>
247     <ns:_ui_vdividedbox width="55%">
248         <ns:_ui_panel_container>
249             <ns:_ui_panel title="Field">
250                 <ns:_ui_list id="fieldList" setState="udx_field" columnCount="1"
                    dragFormat="fieldList" dragEnabled="true" dropEnabled="true"
                    dragMoveEnabled="true" height="100%" width="100%" parentObject=
                    "{this}" change="selectField();"/>
251                 <ns:_ui_btn_bar>
252                     <ns:_ui_btn currentState="saveOrderBtn" click="rankField();" />
253                     <mx:Spacer width="100%"/>
254                     <ns:_ui_btn id="newFieldBtn" currentState="newBtn" click=
                        "fieldForm.newBtn.dispatchEvent(new
                        MouseEvent(MouseEvent.CLICK));"/>
255                     <ns:_ui_btn currentState="refreshBtn" click="getField();" />
```

```
256                 </ns:_ui_btn_bar>
257             </ns:_ui_panel>
258         </ns:_ui_panel_container>
259         <ns:_ui_panel_container height="235">
260             <ns:_ui_panel title="Field Detail">
261                 <ns:udx_field_form id="fieldForm" data="{null}" parentObject=
                    "{this}"/>
262             </ns:_ui_panel>
263         </ns:_ui_panel_container>
264     </ns:_ui_vdividedbox>
265     <ns:states>
266         <mx:State name="section">
267             <mx:SetProperty target="{xmlListContainer}" name="title" value=
                "Section UDX Type"/>
268         </mx:State>
269         <mx:State name="article">
270             <mx:SetProperty target="{xmlListContainer}" name="title" value=
                "Article UDX Type"/>
271         </mx:State>
272         <mx:State name="member">
273             <mx:RemoveChild target="{xmlContainer}"/>
274         </mx:State>
275         <mx:State name="user">
276             <mx:RemoveChild target="{xmlContainer}"/>
277         </mx:State>
278     </ns:states>
279 </ns:_ui_hdividedbox>
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox dataChange="init()" xmlns:mx="http://www.adobe.com/2006/mxml" xmlns:ns="_com.*" width="100%" height="100%">
    <mx:Script>
        <![CDATA[
            import mx.events.FlexEvent;
            import mx.events.ListEvent;
            import mx.events.CloseEvent;
            import mx.controls.Alert;
            import mx.rpc.events.ResultEvent;

            public var parentObject:Object;
            [Bindable]private var Data:Object;
            [Bindable]private var itemExistFlag:Boolean=false;
            //[Bindable]private var app:_app = new _app;

            private function init():void{
                ui();

                Data=null;Data=data;
                if(data == null){
                    this.currentState="no_data";
                }else{
                    this.currentState="view";
                    if(Data.protected=="1"){
                        this.currentState="protected_data";
                    }
                    check();
                }
            }
            private function ui():void{
                if(_app.global["ui_mode"] == "demo"){
                    _app.uiVisibility(saveBtn, false);
                    _app.uiVisibility(removeBtn, false);
                    _app.uiVisibility(saveNewBtn, false);
                }
            }
            private function refreshForm():void{
                this.dispatchEvent(new FlexEvent(FlexEvent.DATA_CHANGE));
            }
            private function check():void{
                t140_name.field.errorString="";
                itemExistFlag=true;

                parentObject.chkXML(formParam, resultHandler);
                function resultHandler(event:ResultEvent):void{
                    if(_app.resultStatus(event,"none")){
                        t140_name.field.errorString="Name already exist.";
                        itemExistFlag=true;
                    }
                    else{
                        itemExistFlag=false;
                        if(!validation()){return}
                    }
                }
            }
```

```
56              private function newForm():void{
57                  parentObject.xmlList.selectedIndex = -1;
58                  parentObject.xmlList.dispatchEvent(new ListEvent(ListEvent.CHANGE));
59                  currentState='new';
60                  Data=null;
61                  ui();
62              }
63              private function add():void{
64                  if(!validation()){return}
65
66                  parentObject.addXML(formParam);
67                  refreshForm();
68              }
69              private function update():void{
70                  if(!validation()){return}
71
72                  parentObject.updateXML(formParam);
73                  refreshForm();
74              }
75              private function remove():void{
76                  _app.confirm("Are you sure you want to remove type?",
                    confirmHandler);
77
78                  function confirmHandler(event:CloseEvent):void{
79                      if (event.detail == Alert.YES) {
80                          parentObject.removeXML(formParam);
81                          refreshForm();
82
83                      }
84                  }
85              }
86              private function validation():Boolean{
87                  return _app.validation(viewForm);
88              }
89          ]]>
90      </mx:Script>
91      <mx:Model id="formParam">
92          <data>
93              <t140_xmlid>{Data.t140_xmlid}</t140_xmlid>
94              <t140_name>{t140_name.data}</t140_name>
95              <t140_desc>{t140_desc.data}</t140_desc>
96              <t140_type>{Data.t140_type}</t140_type>
97          </data>
98      </mx:Model>
99      <mx:Canvas enabled="{viewForm.enabled}" height="100%" width="100%" id=
        "formContainer">
100         <ns:_form id="viewForm">
101             <ns:_form_textInput label="Name" id="t140_name" data="{Data.t140_name}"
                 required="true" change="check();"/>
102             <ns:_form_textInput label="Description" id="t140_desc" data=
                "{Data.t140_desc}"/>
103         </ns:_form>
104     </mx:Canvas>
105     <mx:states>
106         <mx:State name="view">
107             <mx:AddChild position="lastChild">
```

```
108                 <ns:_ui_btn_bar id="viewBtnBar">
109                     <ns:_ui_btn id="newBtn" currentState="newBtn" click=
                        "newForm(); " visible="false" includeInLayout="false"/>
110                     <ns:_ui_btn id="saveBtn" currentState="saveBtn" click=
                        "update();" enabled="{!itemExistFlag}" />
111                     <ns:_ui_btn id="removeBtn" currentState="removeBtn" click=
                        "remove();" />
112                     <ns:_ui_btn id="resetBtn" currentState="resetBtn" click=
                        "refreshForm();" />
113                 </ns:_ui_btn_bar>
114             </mx:AddChild>
115             <mx:AddChild relativeTo="{viewForm}" position="lastChild">
116                 <ns:_form_datetime label="Created On" data="{Data.t140_dt_created}"
                     currentState="read_only"/>
117             </mx:AddChild>
118             <mx:SetProperty target="{t140_name.field}" name="errorString" value=""/>
119         </mx:State>
120         <mx:State name="new">
121             <mx:AddChild position="lastChild">
122                 <ns:_ui_btn_bar>
123                     <ns:_ui_btn id="saveNewBtn" currentState="saveBtn" click=
                        "add();" enabled="{!itemExistFlag}" />
124                     <ns:_ui_btn currentState="cancelBtn" click="refreshForm();" />
125                 </ns:_ui_btn_bar>
126             </mx:AddChild>
127             <mx:SetProperty target="{t140_name.field}" name="errorString" value=""/>
128         </mx:State>
129         <mx:State name="no_data" basedOn="view">
130             <mx:RemoveChild target="{viewBtnBar}"/>
131             <mx:RemoveChild target="{formContainer}"/>
132             <mx:AddChild position="lastChild">
133                 <ns:_ui_note currentState="no_type"/>
134             </mx:AddChild>
135         </mx:State>
136         <mx:State name="protected_data" basedOn="view">
137             <mx:SetProperty target="{t140_name}" name="editable" value="false"/>
138             <mx:SetProperty target="{t140_desc}" name="editable" value="false"/>
139             <mx:SetProperty target="{resetBtn}" name="enabled" value="false"/>
140             <mx:SetProperty target="{removeBtn}" name="enabled" value="false"/>
141         </mx:State>
142     </mx:states>
143 </mx:VBox>
144
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox dataChange="init()" xmlns:mx="http://www.adobe.com/2006/mxml" xmlns:ns="_com.*" width="100%" height="100%">
    <mx:Script>
        <![CDATA[
            import mx.core.UIComponent;
            import mx.events.FlexEvent;
            import mx.events.ListEvent;
            import mx.events.CloseEvent;
            import mx.controls.Alert;
            import mx.rpc.events.ResultEvent;
            import mx.utils.StringUtil;

            public var parentObject:Object;
            [Bindable]private var Data:Object;
            [Bindable]private var itemExistFlag:Boolean=false;
            //[Bindable]private var app:_app = new _app;

            private function init():void{
                ui();

                Data=null;Data=data;
                if(data == null){
                    this.currentState="no_data";
                }else{
                    loadInfo();
                    this.currentState="view";
                    if(Data.protected=="1"){
                        this.currentState="protected_data";
                    }
                    check();
                }
            }
            private function ui():void{
                if(_app.global["ui_mode"] == "demo"){
                    _app.uiVisibility(saveBtn, false);
                    _app.uiVisibility(removeBtn, false);
                    _app.uiVisibility(saveNewBtn, false);
                }
            }
            private function loadInfo():void{
                try{
                    t141_type.selectedIndex = 0;
                    var i:Number = 0;
                    for each(var cat:Object in t141_type.dataProvider){
                        if(cat.iname ==  Data.t141_type){
                            t141_type.selectedIndex = i;
                        }
                        i++;
                    }
                    typeSwitch();
                }catch(error:Error){}
            }
            private function refreshForm():void{
                this.dispatchEvent(new FlexEvent(FlexEvent.DATA_CHANGE));
            }
```

```
56              private function check():void{
57                  t141_label.field.errorString="";
58                  itemExistFlag=true;
59
60                  parentObject.chkField(formParam, resultHandler);
61                  function resultHandler(event:ResultEvent):void{
62                      if(_app.resultStatus(event,"none")){
63                          t141_label.field.errorString="Label already exist.";
64                          itemExistFlag=true;
65                      }
66                      else{
67                          itemExistFlag=false;
68                          if(!validation()){return}
69                      }
70                  }
71              }
72              private function newForm():void{
73                  parentObject.fieldList.selectedIndex = -1;
74                  parentObject.fieldList.dispatchEvent(new ListEvent(ListEvent.CHANGE
                    ));
75                  currentState='new';
76                  Data=null;
77                  typeSwitch();
78                  ui();
79              }
80              private function add():void{
81                  if(!validation()){return}
82
83                  parentObject.addField(formParam);
84                  refreshForm();
85              }
86              private function update():void{
87                  if(!validation()){return}
88
89                  parentObject.updateField(formParam);
90                  refreshForm();
91              }
92              private function remove():void{
93                  _app.confirm("Are you sure you want to remove field?",
                    confirmHandler);
94
95                  function confirmHandler(event:CloseEvent):void{
96                      if (event.detail == Alert.YES) {
97                          parentObject.removeField(formParam);
98                          refreshForm();
99
100                     }
101                 }
102             }
103             private function validation():Boolean{
104                 return _app.validation(viewForm);
105             }
106             private function typeSwitch():void{
107                 var formItem:_form_item;
108                 var targetIndex:Number = viewForm.getChildIndex(viewForm.
                    getChildByName("dynamic"));
```

```
109                 var defaultObject:Object = new Object;
110                 viewForm.removeChild(viewForm.getChildByName("dynamic"));
111 
112                 //reset useSystemDate
113                 _app.uiVisibility(useSystemDate, false);
114                 useSystemDate.data = false;
115 
116                 if(data == null){
117                     defaultObject = "";
118                 }
119                 else{
120                     defaultObject = data.t141_default;
121                 }
122 
123                 t141_remark.toolTip = "";
124 
125                 switch(t141_type.selectedItem.iname){
126                     case "hidden":
127                         var hidden:_form_textArea = new _form_textArea;
128                         hidden.visible = true;
129                         hidden.includeInLayout =true;
130                         formItem = hidden;
131                         hidden.data = defaultObject;
132                     break;
133                     case "number":
134                         var number:_form_textInput = new _form_textInput;
135                         number.numberOnly = true;
136                         number.allowDot = true;
137                         formItem = number;
138                         number.data = defaultObject;
139                     break;
140                     case "string":
141                         var string:_form_textInput = new _form_textInput;
142                         formItem = string;
143                         string.data = defaultObject;
144                     break;
145                     case "browse":
146                         var browse:_form_textInput = new _form_textInput;
147                         formItem = browse;
148                         browse.data = defaultObject;
149                         browse.browse = true;
150                     break;
151                     case "link":
152                         var link:_form_textInput = new _form_textInput;
153                         formItem = link;
154                         link.data = defaultObject;
155                         link.browseLink = true;
156                     break;
157                     case "date":
158                         var date:_form_datetime = new _form_datetime;
159                         formItem = date;
160                         date.data = _app.decodeDate(defaultObject.toString(), true);
161                         date.minDate =new Date(0);
162                         date.currentState = "date_only";
163                         applyUseSystemDate(date, defaultObject.toString());
164                     break;
```

```
165                     case "datetime":
166                         var datetime:_form_datetime = new _form_datetime;
167                         formItem = datetime;
168                         datetime.data = _app.decodeDate(defaultObject.toString(),
                            true);
169                         datetime.minDate =new Date(0);
170                         applyUseSystemDate(datetime, defaultObject.toString());
171                     break;
172                     case "checkbox":
173                         var checkbox:_form_checkBox_standalone = new
                            _form_checkBox_standalone;
174                         formItem = checkbox;
175                         checkbox.fieldLabel = "Selected";
176                         checkbox.data = Boolean(int(defaultObject));
177                     break;
178                     case "textarea":
179                         var textarea:_form_textArea = new _form_textArea;
180                         formItem = textarea;
181                         textarea.data = defaultObject;
182                     break;
183                     case "combobox":
184                         var combobox:_form_comboBox = new _form_comboBox;
185                         formItem = combobox;
186                         combobox.toolTip = "Modify selections in Remark";
187                         if(Boolean(int(t141_required.data))){
188                             try{
189                                 combobox.dataProvider = _app.generateComboBoxSource
                                    (t141_remark.data.toString());
190                             }
191                             catch(error:Error){
192                                 combobox.dataProvider = _app.generateComboBoxSource
                                    ("");
193                             }
194                         }
195                         else{
196                             try{
197                                 combobox.dataProvider = _app.getNAObject(_app.
                                    generateComboBoxSource(t141_remark.data.toString
                                    ()), "data", "label", "");
198                             }
199                             catch(error:Error){
200                                 combobox.dataProvider = _app.getNAObject(_app.
                                    generateComboBoxSource(""), "data", "label", "");
201                             }
202                         }
203                         combobox.data = defaultObject;
204 
205                         t141_remark.toolTip = "Format
                            :\n\n<value>|<label>;\n<value>|<label>;\n<value>|<label>;\n\
                            nExample :\n\nmale|Male;\nfemale|Female;"
206                     break;
207                 }
208                 formItem.label = "Default Value";
209                 formItem.name = "dynamic";
210                 formItem.id = "t141_default";
211                 formItem.addEventListener(Event.CHANGE, updateData);
```

```
212                 viewForm.addChildAt(formItem, targetIndex);
213                 formItem.dispatchEvent(new Event(Event.CHANGE));
214             }
215             private function updateData(event:Event):void{
216                 switch(t141_type.selectedItem.iname){
217                     case "date":
218                         formParam.t141_default = _app.encodeDateXML(event.
                            currentTarget.fieldData);
219                     break;
220                     case "datetime":
221                         formParam.t141_default = _app.encodeDateXML(event.
                            currentTarget.fieldData);
222                     break;
223                     case "combobox":
224                         formParam.t141_default = event.currentTarget.data;
225                     break;
226                     default:
227                         formParam.t141_default = event.currentTarget.data.toString
                            ();
228                     break;
229 
230                 }
231                 if(useSystemDate.data){
232                     formParam.t141_default = "";
233                 }
234                 event.currentTarget.enabled = !useSystemDate.data;
235             }
236             private function applyUseSystemDate(objectIn:_form_datetime, stringIn:
                String):void{
237                 _app.uiVisibility(useSystemDate, true);
238                 if(StringUtil.trim(stringIn) == ""){
239                     useSystemDate.data = true;
240                 }
241                 else{
242                     useSystemDate.data = false;
243                 }
244             }
245         ]]>
246     </mx:Script>
247     <mx:Model id="formParam">
248         <data>
249             <t141_structid>{Data.t141_structid}</t141_structid>
250             <t141_xmlid>{Data.t141_xmlid}</t141_xmlid>
251             <t141_label>{t141_label.data}</t141_label>
252             <t141_type>{t141_type.selectedItem.iname}</t141_type>
253             <t141_required>{t141_required.data}</t141_required>
254             <t141_remark>{t141_remark.data}</t141_remark>
255             <t141_default></t141_default>
256         </data>
257     </mx:Model>
258     <mx:Canvas enabled="{viewForm.enabled}" height="100%" width="100%" id=
        "formContainer">
259         <ns:_form id="viewForm">
260             <ns:_form_textInput label="Name" id="t141_label" data=
                "{Data.t141_label}" required="true" change="check();" wordOnly="false"
                allowSpace="true"/>
```

```
261             <ns:_form_item>
262                 <ns:_form_checkBox_container>
263                     <ns:_form_checkBox id="t141_required" label="" fieldLabel=
                        "Required" data="{Data.t141_required}" change="typeSwitch();"/>
264                 </ns:_form_checkBox_container>
265             </ns:_form_item>
266             <mx:HRule id="hrule" width="100%"/>
267             <mx:Spacer height="3"/>
268             <ns:_form_item label="Type" direction="horizontal" horizontalGap="0">
269                 <mx:ComboBox id="t141_type" labelField="idesc" change=
                    "typeSwitch();">
270                     <mx:ArrayCollection>
271                         <mx:Object idesc="Text" ikey="1" iname="string"/>
272                         <mx:Object idesc="Text Area" ikey="7" iname="textarea"/>
273                         <mx:Object idesc="Number" ikey="9" iname="number"/>
274                         <mx:Object idesc="Browse" ikey="2" iname="browse"/>
275                         <mx:Object idesc="Browse Link" ikey="8" iname="link"/>
276                         <mx:Object idesc="Date" ikey="3" iname="date"/>
277                         <mx:Object idesc="Date and Time" ikey="4" iname="datetime"/>
278                         <mx:Object idesc="Check Box" ikey="5" iname="checkbox"/>
279                         <mx:Object idesc="Combo Box" ikey="6" iname="combobox"/>
280                         <mx:Object idesc="Hidden" ikey="10" iname="hidden"/>
281                     </mx:ArrayCollection>
282                 </mx:ComboBox>
283             </ns:_form_item>
284             <ns:_form_textArea label="Remark" id="t141_remark" data=
                "{Data.t141_remark}" change="typeSwitch();" visible=
                "{(t141_type.selectedItem.iname == 'combobox')}" includeInLayout=
                "{(t141_type.selectedItem.iname == 'combobox')}"/>
285             <ns:_form_textInput label="Default Value" id="t141_default" name=
                "dynamic" data="{Data.t141_default}"/>
286             <ns:_form_item>
287                 <ns:_form_checkBox_container>
288                     <ns:_form_checkBox id="useSystemDate" label="" fieldLabel="Use
                        System Date" change=
                        "_form_item(viewForm.getChildByName('dynamic')).dispatchEvent(ne
                        w Event(Event.CHANGE));"/>
289                 </ns:_form_checkBox_container>
290             </ns:_form_item>
291         </ns:_form>
292     </mx:Canvas>
293     <mx:states>
294         <mx:State name="view">
295             <mx:AddChild position="lastChild">
296                 <ns:_ui_btn_bar id="viewBtnBar">
297                     <ns:_ui_btn id="newBtn" currentState="newBtn" click=
                        "newForm(); " visible="false" includeInLayout="false"/>
298                     <ns:_ui_btn id="saveBtn" currentState="saveBtn" click=
                        "update();" enabled="{!itemExistFlag}" />
299                     <ns:_ui_btn id="removeBtn" currentState="removeBtn" click=
                        "remove();" />
300                     <ns:_ui_btn id="resetBtn" currentState="resetBtn" click=
                        "refreshForm();" />
301                 </ns:_ui_btn_bar>
302             </mx:AddChild>
303             <mx:AddChild relativeTo="{hrule}" position="before">
```

```
304                 <ns:_form_datetime label="Created On" data="{Data.t141_dt_created}"
                     currentState="read_only"/>
305             </mx:AddChild>
306             <mx:SetProperty target="{t141_label.field}" name="errorString" value=""
                />
307         </mx:State>
308         <mx:State name="new">
309             <mx:AddChild position="lastChild">
310                 <ns:_ui_btn_bar>
311                     <ns:_ui_btn id="saveNewBtn" currentState="saveBtn" click=
                        "add();" enabled="{!itemExistFlag}" />
312                     <ns:_ui_btn currentState="cancelBtn" click="refreshForm();" />
313                 </ns:_ui_btn_bar>
314             </mx:AddChild>
315             <mx:SetProperty target="{t141_label.field}" name="errorString" value=""
                />
316         </mx:State>
317         <mx:State name="no_data" basedOn="view">
318             <mx:RemoveChild target="{viewBtnBar}"/>
319             <mx:RemoveChild target="{formContainer}"/>
320             <mx:AddChild position="lastChild">
321                 <ns:_ui_note currentState="no_field"/>
322             </mx:AddChild>
323         </mx:State>
324         <mx:State name="protected_data" basedOn="view">
325             <mx:SetProperty target="{resetBtn}" name="enabled" value="false"/>
326             <mx:SetProperty target="{removeBtn}" name="enabled" value="false"/>
327         </mx:State>
328     </mx:states>
329 </mx:VBox>
330
```

```
<?xml version="1.0" encoding="utf-8"?>
<ns:_form creationComplete="init();" dataChange="init();dataChange();" xmlns:mx=
"http://www.adobe.com/2006/mxml" xmlns:ns="_com.*" width="100%">
    <mx:Script>
        <![CDATA[
            import mx.events.FlexEvent;
            import mx.containers.FormItem;
            import mx.utils.StringUtil;

            [Bindable]public var fieldData:Object;
            [Bindable]public var extendedData:Object;
            public var xmlEditor:Boolean = true;

            private function init():void{
                if(!_app.access("MOD0299") || xmlEditor != true){
                    //this.currentState="normal";
                }else{
                    //this.currentState="";
                }
            }
            private function dataChange():void{
                //fieldData = data;
                var formItem:_form_item;
                this.removeAllChildren();
                try{
                    for each(var field:XML in XMLList(data)){
                        switch(String(field.t141_type)){
                            case "hidden":
                                var hidden:_form_textArea = new _form_textArea;
                                hidden.visible = false;
                                hidden.includeInLayout =false;
                                formItem = hidden;
                                hidden.data = field.t141_default;
                            break;
                            case "number":
                                var number:_form_textInput = new _form_textInput;
                                number.numberOnly = true;
                                number.allowDot = true;
                                formItem = number;
                                number.data = field.t141_default;
                            break;
                            case "string":
                                var string:_form_textInput = new _form_textInput;
                                formItem = string;
                                string.data = field.t141_default;
                            break;
                            case "browse":
                                var browse:_form_textInput = new _form_textInput;
                                formItem = browse;
                                browse.browse = true;
                                browse.data = field.t141_default;
                            break;
                            case "link":
                                var link:_form_textInput = new _form_textInput;
                                formItem = link;
                                link.data = field.t141_default;
```

```
56                              link.browseLink = true;
57                          break;
58                          case "date":
59                              var date:_form_datetime = new _form_datetime;
60                              formItem = date;
61                              date.data = _app.decodeDate(field.t141_default,
                                true);
62                              date.minDate =new Date(0);
63                              date.currentState = "date_only";
64                          break;
65                          case "datetime":
66                              var datetime:_form_datetime = new _form_datetime;
67                              formItem = datetime;
68                              datetime.data = _app.decodeDate(field.t141_default,
                                 true);
69                              datetime.minDate =new Date(0);
70                          break;
71                          case "checkbox":
72                              var checkbox:_form_checkBox_standalone = new
                                _form_checkBox_standalone;
73                              formItem = checkbox;
74                              checkbox.data = Boolean(int(field.t141_default));
75                          break;
76                          case "textarea":
77                              var textarea:_form_textArea = new _form_textArea;
78                              formItem = textarea;
79                              textarea.data = field.t141_default;
80                          break;
81                          case "combobox":
82                              var combobox:_form_comboBox = new _form_comboBox;
83                              formItem = combobox;
84                              if(Boolean(int(field.t141_required))){
85                                  combobox.dataProvider = _app.
                                    generateComboBoxSource(field.t141_remark);
86                              }
87                              else{
88                                  combobox.dataProvider = _app.getNAObject(_app.
                                    generateComboBoxSource(field.t141_remark),
                                    "data", "label", "");
89                              }
90                              combobox.data = field.t141_default;
91                          break;
92                      }
93                      formItem.style = "system";
94                      formItem.label = field.t141_label;
95                      formItem.name = field.t141_structid;
96                      formItem.remark = field.t141_type;
97                      formItem.required = Boolean(int(field.t141_required));
98                      formItem.addEventListener(Event.CHANGE, updateData);
99                      this.addChild(formItem);
100                     formItem.dispatchEvent(new Event(Event.CHANGE));
101                 }
102             }catch(error:Error){}
103             extendedDataChange();
104         }
105         public function validation():Boolean{
```

```
106                 return _app.validation(this);
107             }
108             public function extendedDataChange():void{
109                 var formItem:Object;
110                 for each(formItem in this.getChildren()){
111                     formItem.data = "";
112                 }
113                 try{
114                     for each(var field:XML in XMLList(XML(extendedData).children
                        ())){
115                         formItem = this.getChildByName(field.t141_structid);
116                         switch(formItem.remark){
117                             case "date":
118                                 formItem.data = _app.decodeDate(XML(field.data));
119                             break;
120                             case "datetime":
121                                 formItem.data = _app.decodeDate(XML(field.data));
122                             break;
123                             case "combobox":
124                                 formItem.data = field.data;
125                             break;
126                             default:
127                                 if(field.data.data.toString() != ""){
128                                     formItem.data = field.data.data.toString();
129                                 }
130                                 else{
131                                     formItem.data = field.data.toString();
132                                 }
133                             break;
134                         }
135                     }
136                     for each(formItem in this.getChildren()){
137                         switch(formItem.remark){
138                             case "date":
139                                 formItem.fieldData = formItem.data;
140                             break;
141                             case "datetime":
142                                 formItem.fieldData = formItem.data;
143                             break;
144                         }
145                     }
146                 }catch(error:Error){}
147                 updateData(new Event(Event.CHANGE));
148             }
149             private function updateData(event:Event):void{
150                 //data.children().(label==event.currentTarget.toolTip).data =
                    event.currentTarget.text;
151 
                    //Alert.show(data.children().(label==event.currentTarget.toolTip).da
                    ta);
152                 var tempString:String = "";
153                 for each(var formItem:Object in this.getChildren()){
154                             //_app.error(formItem.data.toString());
155                     tempString += "<field>";
156                     tempString += "<t141_structid>" + formItem.name +
                        "</t141_structid>";
```

```
157                     switch(formItem.remark){
158                         case "date":
159                             tempString += "<data>" + _app.encodeDateXML(formItem.
                                fieldData) + "</data>";
160                         break;
161                         case "datetime":
162                             tempString += "<data>" + _app.encodeDateXML(formItem.
                                fieldData) + "</data>";
163                         break;
164                         case "hidden":
165                             tempString += "<data>" + formItem.data + "</data>";
166                         break;
167                         case "XML":
168                             tempString += "<data>" + formItem.data + "</data>";
169                         break;
170                         default:
171                             tempString += "<data>" + _app.CDATA(formItem.data).
                                toXMLString() + "</data>";
172                         break;
173                     }
174                     tempString += "</field>";
175                 }
176                 fieldData = tempString;
177                 //_app.error(fieldData.toString());
178                 //7mx.controls.Alert.show(tempString);
179             }
180         ]]>
181     </mx:Script>
182 </ns:_form>
183
```

```
<?xml version="1.0" encoding="utf-8"?>
<ns:_ui_hdividedbox creationComplete="init();ui();" dataChange="init()" xmlns:mx="http://www.adobe.com/2006/mxml" xmlns:ns="_com.*">
    <mx:Script>
        <![CDATA[
            import mx.controls.Alert;
            import mx.events.ListEvent;
            import mx.rpc.events.ResultEvent;
            import mx.events.FlexEvent;
            import mx.collections.HierarchicalData;

            private var HTTPService:_sys_httpService;
            public var parentObject:Object;
            private var selectedId:Number=-1;

            private function init(newId:String="-1"):void{
                currentState = "";
                switch(String(data)){
                    case "approval":
                        currentState = "approval";
                        getAllArticles(newId);
                    break;
                    case "":
                    break;
                    case "search":
                        getAllArticles(newId, false);
                    break;
                    default:
                        getAllArticles(newId);
                    break;
                }
            }
            private function ui():void{
                if(_app.global["ui_mode"] == "fe"){
                    try{
                        articleAccordian.removeChild(articleAttached);
                    }catch(error:Error){}
                }
                if(_app.global["linkSelect"] == true || parent.toString().split(".")[1] == "moduleLoader"){
                    try{
                        articleAccordian.removeChild(articleStatistic);
                    }catch(error:Error){}
                }
                _app.uiVisibility(newBtn, _app.access("MOD0261"));
            }
            private function resetData():void{
                articleForm.data2 = null;
                articleForm.data = null;
                articleStatistic.data = null;
                articleAttached.data = null;
            }
            public function getAllArticles(newId:String = "", manualTrigger:Boolean=true):void{
                //resetData();
                //try{
```

```
54                      articleList.dataProvider = new HierarchicalData();
55                      if(!manualTrigger){
56                          listCount.text = "";
57                          articleList.selectedIndex=-1;
58                          articleList.dispatchEvent(new ListEvent(ListEvent.CHANGE));
59                          return;
60                      }
61
62                      var HTTPparams:Object = new Object;
63                      HTTPparams.func="cArticle.get_articles";
64                      HTTPparams.searchstr= searchBar.searchString;
65                      //Alert.show(data.toString());
66                      switch(String(data)){
67                          case "approval":
68                              HTTPparams.func="cArticle.get_pending";
69                              HTTPparams.allpending = showAllPending.selectedItem.
                                data;
70                          break;
71                          case "search":
72                              HTTPparams.searchall= "1";
73                          break;
74                          case "nonattach":
75                              HTTPparams.t40_sectionid="";
76                          break;
77                          case "root":
78                              return;
79                          break;
80                          default:
81                              HTTPparams.t40_sectionid=data;
82                          break;
83                      }
84                      HTTPService = new _sys_httpService(_app.servicesPath, null,
                        HTTPparams, resultHandler);
85                  //}catch(error:Error){}
86                  //Alert.show(data.toString());
87                  function resultHandler(event:ResultEvent):void{
88                      //Alert.show(event.result.toString());
89                      if(!_app.resultStatus(event)){return;}
90
91                      if(_app.resultReturn(event) != ""){
92                          parentObject.articleContainer.label = "Article [" + _app.
                            resultReturn(event) + "]";
93                      }else{
94                          parentObject.articleContainer.label = "Article";
95                      }
96
97                      try{
98                          articleList.dataProvider = new HierarchicalData(XML(event.
                            result).content.data);
99                          listCount.text = "Total Articles: " + XML(event.result).
                            content.data.length();
100                         var i:Number=-1;
101                         for each(var article:XML in XML(event.result).content.data){
102                             i++;
103                             for each(var version:XML in article.children()){
104                                 i++;
```

```
105                             if(version.@t51_versionid==newId){
106
                                    //mx.controls.Alert.show(version.@t51_versionid
                                    + " " + newId);
107                                 selectedId = i;
108                             }
109                         }
110                     }
111                     if(data != "search"){
112                         articleList.callLater(articleList.expandAll);
113                         articleList.callLater(defaultListIndex);
114                     }
115                 }catch(error:Error){}
116             }
117         }
118         private function defaultListIndex():void{
119             //mx.controls.Alert.show(selectedId.toString());
120             articleList.selectedIndex=Number(selectedId);
121             articleList.dispatchEvent(new ListEvent(ListEvent.CHANGE));
122         }
123         private function getArticle():void{
124             resetData();
125
126             selectedId = articleList.selectedIndex;
127             if(!articleList.selectedItems.length > 0){return;}
128             //articleStatistic.data = "dummy";
129             if(articleList.selectedItem.@t51_versionid != ""){
130                 articleStatistic.targetField = "t90_versionid";
131                 articleStatistic.data = articleList.selectedItem.@t51_versionid;
132                 getVersion();
133             }
134             else{
135                 articleStatistic.targetField = "t90_articleid";
136                 articleStatistic.data = articleList.selectedItem.@t50_articleid;
137             }
138             articleAttached.data = articleList.selectedItem.@t50_articleid;
139             //Alert.show( data + "/n" +
                articleList.selectedItem.@t50_articleid + "/n" +
                articleList.selectedItem.@t51_versionid);
140             var HTTPparams:Object = new Object;
141             HTTPparams.func="cArticle.get_article_info";
142             HTTPparams.t40_sectionid = data;
143             HTTPparams.t50_articleid = articleList.selectedItem.@t50_articleid;
144             HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, resultHandler);
145
146             function resultHandler(event:ResultEvent):void{
147                 if(!_app.resultStatus(event)){return;}
148
149                 var newArr:Array=_app.getContentData(event);
150
151                 //mx.controls.Alert.show(event.result.toString());
152                 articleForm.data2 =newArr[0];
153                 articleForm.dispatchEvent(new FlexEvent(FlexEvent.DATA_CHANGE));
154
155                 //articleAttached.data=newArr[0];
```

```
156                     //articleStatistic.data=newArr[0];
157
158                     if(articleList.selectedItem.@t51_versionid != ""){
159                         //getVersion();
160                     }
161                 }
162             }
163             public function addArticle(formParam:Object):void{
164                 var HTTPparams:Object = new Object;
165                 HTTPparams = formParam;
166                 //mx.controls.Alert.show(data.toString());
167                 HTTPparams.func="cArticle.create_article";
168                 if(data == "nonattach" || data == "search"){
169                     HTTPparams.t60_sectionid = "";
170                 }
171                 else{
172                     HTTPparams.t60_sectionid = data;
173                 }
174                 HTTPparams.t51_dt_pub=_app.encodeDate(formParam.t51_dt_pub);
175                 HTTPparams.t51_dt_exp=_app.encodeDate(formParam.t51_dt_exp);
176                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, articleResultHandler, null, null, true, true);
177             }
178             public function setArticle(formParam:Object, updateVersion:Boolean,
                newVersion:Boolean):void{
179                 var HTTPparams:Object = new Object;
180                 HTTPparams = formParam;
181                 HTTPparams.func="cArticle.update_article";
182                 HTTPparams.update_version = int(updateVersion);
183                 HTTPparams.v_new_version = int(newVersion);
184                 HTTPparams.t51_dt_pub=_app.encodeDate(formParam.t51_dt_pub);
185                 HTTPparams.t51_dt_exp=_app.encodeDate(formParam.t51_dt_exp);
186                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, articleResultHandler, null, null, true, true);
187
188                 //_app.error(formParam.t51_dt_pub.toString());
189             }
190             public function removeArticle(formParam:Object):void{
191                 var HTTPparams:Object = new Object;
192                 HTTPparams = formParam;
193                 HTTPparams.func="cArticle.delete_article";
194                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, articleResultHandler, null, null, true, true);
195             }
196             private function articleResultHandler(event:ResultEvent):void{
197                 //_app.error(event.result.toString());
198                 if(!_app.resultStatus(event)){return;}
199                 init(_app.resultReturn(event));
200             }
201             private function getVersion():void{
202                 var HTTPparams:Object = new Object;
203                 HTTPparams.func="cArticle.get_version_info";
204                 HTTPparams.t51_versionid = articleList.selectedItem.@t51_versionid;
205                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
206
```

```
207                 function resultHandler(event:ResultEvent):void{
208                     //_app.error(event.result.toString());
209                     //_app.error(event.result.toString());
210                     if(!_app.resultStatus(event)){
211                         articleForm.data = new Object;
212                         return;
213                     }
214 
215                     var newArr:Array=_app.getContentData(event);
216 
217                     //mx.controls.Alert.show(event.result.toString());
218                     articleForm.data =newArr[0];
219                     //articleAttached.data=newArr[0];
220                     //articleStatistic.data=newArr[0];
221                 }
222             }
223             public function removeVersion(formParam:Object):void{
224                 var HTTPparams:Object = new Object;
225                 HTTPparams = formParam;
226                 HTTPparams.func="cArticle.delete_version";
227                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, articleResultHandler, null, null, true, true);
228             }
229             public function approval(formParam:Object, approve:Boolean):void{
230                 var HTTPparams:Object = new Object;
231                 HTTPparams = formParam;
232                 HTTPparams.func="cArticle.approve_article";
233                 HTTPparams.t51_app_status=int(approve);
234                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, articleResultHandler, null, null, true, true);
235             }
236             private function articleListLabel(item:Object, column:
                AdvancedDataGridColumn):String{
237                 var tempString:String;
238                 if(item.@t50_title == ""){
239                     //tempString = "Version " + item.@t51_versioncode;
240                 }
241                 else{
242                     tempString = item.@t50_title;
243                 }
244                 return tempString;
245             }
246         ]]>
247     </mx:Script>
248 
249     <ns:_ui_vdividedbox width="50%">
250         <ns:_ui_panel_container height="60%" creationPolicy="all">
251             <ns:_ui_panel title="Article">
252                 <ns:_ui_search id="searchBar" currentState="search" searchFunction=
                    "{getAllArticles}"/>
253                 <ns:_ui_advancedDatagrid id="articleList" change="getArticle();"
                    parentObject="{this}" editable="false" styleName="article">
254                     <ns:columns>
255                         <mx:AdvancedDataGridColumn id="t50_title" dataField=
                            "@t50_title" headerText="Title" labelFunction=
                            "articleListLabel"/>
```

```
256                         <mx:AdvancedDataGridColumn id="t51_versioncode" dataField=
                            "@t51_versioncode" headerText="Version" width="38"/>
257                         <mx:AdvancedDataGridColumn id="dt_created" dataField=
                            "@dt_created" headerText="Created" width="60"/>
258                     </ns:columns>
259                     <ns:rendererProviders>
260                          <mx:AdvancedDataGridRendererProvider column=
                             "{t51_versioncode}" renderer="mx.controls.Label"/>
261                          <mx:AdvancedDataGridRendererProvider column="{dt_created}"
                              renderer="mx.controls.Label"/>
262                     </ns:rendererProviders>
263                 </ns:_ui_advancedDatagrid>
264                 <ns:_ui_btn_bar id="controlbar1">
265                     <mx:Label id="listCount" fontWeight="bold"/>
266                     <mx:Spacer width="100%"/>
267                     <mx:ComboBox id="showAllPending" change="getAllArticles();"
                        visible="false" includeInLayout="false">
268                         <mx:Array>
269                             <mx:Object label="Articles Appointed to You" data="0"/>
270                             <mx:Object label="All Pending Articles" data="1"/>
271                         </mx:Array>
272                     </mx:ComboBox>
273                     <ns:_ui_btn id="newBtn" currentState="newBtn" click=
                        "articleAccordian.selectedChild
                        =Information;articleForm.newBtn.dispatchEvent(new
                        MouseEvent(MouseEvent.CLICK));"/>
274                     <ns:_ui_btn id="refreshBtn" currentState="refreshBtn" click=
                        "getAllArticles();" />
275                 </ns:_ui_btn_bar>
276             </ns:_ui_panel>
277         </ns:_ui_panel_container>
278     </ns:_ui_vdividedbox>
279     <ns:_ui_panel minWidth="360" width="360" styleName="accordianContainer">
280         <mx:Accordion id="articleAccordian" height="100%" width="100%"
            creationPolicy="all">
281             <ns:_ui_panel_container label="Article Detail" id="Information">
282                 <ns:_ui_panel styleName="accordianPanel">
283                     <ns:pub_art_form id="articleForm" data="{data}" parentObject=
                        "{this}" />
284                 </ns:_ui_panel>
285             </ns:_ui_panel_container>
286             <ns:pub_art_attach id="articleAttached" label="Article Publication"
                parentObject="{this}"/>
287             <ns:pub_art_chart id="articleStatistic" label="Article Web Statistic"
                parentObject="{this}"/>
288         </mx:Accordion>
289     </ns:_ui_panel>
290     <ns:states>
291         <mx:State name="approval">
292             <mx:RemoveChild target="{newBtn}"/>
293             <mx:SetProperty target="{showAllPending}" name="visible" value="true"/>
294             <mx:SetProperty target="{showAllPending}" name="includeInLayout" value=
                "true"/>
295         </mx:State>
296     </ns:states>
297 </ns:_ui_hdividedbox>
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox creationComplete="preinit();init();" dataChange="init();" xmlns:mx=
"http://www.adobe.com/2006/mxml" xmlns:ns="_com.*" width="100%" height="100%">
    <mx:Script>
        <![CDATA[
            import mx.events.ListEvent;
            import mx.events.CloseEvent;
            import mx.events.FlexEvent;
            import mx.rpc.events.ResultEvent;
            import mx.controls.Alert;

            private var HTTPService:_sys_httpService;

            public var parentObject:Object;
            [Bindable]public var data2:Object;
            [Bindable]private var Data:Object;
            [Bindable]private var Data2:Object;
            [Bindable]private var itemExistFlag:Boolean=false;
            [Bindable]private var datePubNow:Date;
            [Bindable]private var dateExpNow:Date;
            //[Bindable]private var app:_app = new _app;

            private function preinit():void{
                getEditorTypes();
                getDynamicTypes();
            }
            private function init():void{
                ui();
                datePubNow = new Date(_app.systemDate);
                dateExpNow = new Date(_app.systemDate);
                dateExpNow.setFullYear(dateExpNow.getFullYear()+5);

                Data=null;Data=data;
                Data2=null;Data2=data2;

                _app.global["articleContent"] = "";

                if(data != null && data2 != null){
                    this.currentState="with_version";
                    if(parentObject.currentState == "approval"){
                        this.currentState="with_version_approval";
                    }
                    loadInfo(true);
                    if(Data=="root"){
                        this.currentState="root_data";
                    }
                    ui();
                }
                else if(data == null && data2 != null){
                    loadInfo(false);
                    this.currentState="no_version";
                    if(parentObject.currentState == "approval"){
                        this.currentState="no_version_approval";
                    }
                }else{
                    this.currentState="no_data";
```

```
56                  }
57              }
58              private function ui():void{
59                  if(_app.global["ui_mode"] == "fe"){
60                      _app.uiVisibility(approverSelect, false);
61                      _app.uiVisibility(approverDisplay, false);
62                      _app.uiVisibility(t51_secure, false);
63                      if(!_app.global["unlockUDX"]){
64                          _app.uiVisibility(viewForm3, false);
65                          _app.uiVisibility(t50_xml_typeid_form_item, false);
66                          try{
67                              t50_xml_typeid.selectedIndex == 0;
68                          }catch(error:Error){}
69                      }
70                  }
71
72                  if(_app.global["linkSelect"] == true || parent.toString().split("."
                    )[1] == "moduleLoader"){
73                      _app.uiVisibility(editContentBtn, false);
74                      _app.uiVisibility(selectBtn, true);
75                  }else{
76                      _app.uiVisibility(editContentBtn, true);
77                      _app.uiVisibility(selectBtn, false);
78                  }
79                  _app.uiVisibility(saveBtn, _app.access("MOD0262"));
80                  _app.uiVisibility(saveAsBtn, _app.access("MOD0262"));
81                  _app.uiVisibility(removeBtn, _app.access("MOD0263"));
82                  _app.uiVisibility(approveBtn, _app.access("MOD0264"));
83                  _app.uiVisibility(rejectBtn, _app.access("MOD0264"));
84              }
85              private function getApprover():void{
86                  var HTTPparams:Object = new Object;
87                  HTTPparams.func="cArticle.get_approvers";
88                  HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
89
90                  function resultHandler(event:ResultEvent):void{
91                      if(!_app.resultStatus(event)){return;}
92
93                      var newArr:Array=_app.getContentData(event);
94                      t51_app_by.dataProvider = newArr;
95
96                      try{
97                          var i:Number = 0;
98                          for each(var approver:Object in t51_app_by.dataProvider){
99                              if(approver.t10_userid ==  Data.t51_app_by){
100                                 t51_app_by.selectedIndex = i;
101                             }
102                             i++;
103                         }
104                     }catch(error:Error){}
105                 }
106             }
107             private function getEditorTypes():void{
108                 /*
109                 var HTTPparams:Object = new Object;
```

```
110                 HTTPparams.func="cArticle.get_editor_type";
111                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
112
113                 function resultHandler(event:ResultEvent):void{
114                     if(!_app.resultStatus(event)){return;}
115
116                     var newArr:Array=_app.getContentData(event);
117                     t51_typeid.dataProvider = newArr;
118                 }
119                 */
120             }
121             private function getDynamicTypes():void{
122                 var HTTPparams:Object = new Object;
123                 HTTPparams.func="cArticle.get_xml_type";
124                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
125
126                 function resultHandler(event:ResultEvent):void{
127                     if(!_app.resultStatus(event)){return;}
128
129                     var newArr:Array=_app.getContentData(event);
130                     t50_xml_typeid.dataProvider = _app.getNAObject(newArr,
                        "t140_xmlid", "t140_name");
131                 }
132             }
133             private function loadInfo(version:Boolean):void{
134                 if(version){
135                     getApprover();
136
137                     _app.global["articleContent"] = Data.t51_content;
138                     if(Data.t51_app_status == "1"){
139                         approverDisplay.data = "Approved on " + _app.formatDate(
                            Data.t51_app_dt, "datetime24");
140                     }
141                     else if(Data.t51_app_status == "0"){
142                         approverDisplay.data = "Rejected on " + _app.formatDate(
                            Data.t51_app_dt, "datetime24");
143                     }
144                     else{
145                         approverDisplay.data = "Pending";
146                     }
147                 }
148                 try{
149                     var i:Number = 0;
150                     for each(var cat:Object in t50_xml_typeid.dataProvider){
151                         if(cat.t140_xmlid ==  Data2.t50_xml_typeid){
152                             t50_xml_typeid.selectedIndex = i;
153                         }
154                         i++;
155                     }
156                     t51_xml.extendedDataChange();
157                 }catch(error:Error){}
158                 try{
159                     i = 0;
160                     for each(var editor:Object in t51_typeid.dataProvider){
```

```
161                         if(editor.ikey ==  Data.t51_typeid){
162                             t51_typeid.selectedIndex = i;
163                         }
164                         i++;
165                     }
166                 }catch(error:Error){}
167             }
168             public function refreshForm():void{
169                 this.dispatchEvent(new FlexEvent(FlexEvent.DATA_CHANGE));
170             }
171             private function newForm():void{
172                 parentObject.articleList.selectedIndex = -1;
173                 parentObject.articleList.dispatchEvent(new ListEvent(ListEvent.
                    CHANGE));
174 
175                 _app.global["articleContent"] = "";
176                 currentState='new';
177                 Data2=null;
178                 Data=null;
179                 getApprover();
180             }
181             private function add():void{
182                 if(!validation()){return}
183 
184                 t51_content.data = _app.global["articleContent"];
185                 parentObject.addArticle(formParam);
186                 refreshForm();
187             }
188             private function updateSE(newVersion:Boolean = false):void{
189                 t51_content.data = _app.global["articleContent"];
190                 if(currentState == "no_version"){
191                     if(!_app.validation(viewForm)){return}
192                     parentObject.setArticle(formParam, false, newVersion);
193                 }
194                 else{
195                     if(!validation()){return}
196                     parentObject.setArticle(formParam, true, newVersion);
197                 }
198                 refreshForm();
199             }
200             private function update():void{
201                 if(!validation()){return}
202                 Alert.buttonWidth = 150;
203                 Alert.yesLabel = "Update as New Version";
204                 Alert.noLabel = "Update Current Version";
205                 Alert.show("Please select an option", "Confirmation Needed", Alert.
                    YES|Alert.NO, null, confirmHandler, _app.exclamationIcon,Alert.YES);
206                 Alert.buttonWidth = 65;
207                 Alert.yesLabel = "Yes";
208                 Alert.noLabel = "No";
209 
210                 t51_content.data = _app.global["articleContent"];
211                 function confirmHandler(event:CloseEvent):void{
212                     if (event.detail == Alert.YES) {
213                         parentObject.setArticle(formParam, true);
214                     }
```

```
215                     else{
216                         parentObject.setArticle(formParam, false);
217                     }
218                     refreshForm();
219                 }
220             }
221             private function remove():void{
222                 if(Data == null){
223                     _app.confirm("Are you sure you want to remove article?",
                    confirmHandler);
224                 }
225                 else{
226                     _app.confirm("Are you sure you want to remove version?",
                    confirmHandler);
227                 }
228                 function confirmHandler(event:CloseEvent):void{
229                     if (event.detail == Alert.YES) {
230                         if(Data == null){
231                             parentObject.removeArticle(formParam);
232                         }
233                         else{
234                             parentObject.removeVersion(formParam);
235                         }
236                         refreshForm();
237                     }
238                 }
239             }
240             private function validation():Boolean{
241                 return (_app.validation(viewForm) && t51_xml.validation());
242             }
243             private function url(actionIn:String):void{
244                 var url:String = _app.generateURL(Data2, Data);
245 
246                 switch(actionIn){
247                     case "select":
248                         _app.SelectFile(url);
249                     break;
250                     case "view":
251                         _app.loadPreview(url);
252                     break;
253                 }
254             }
255         ]]>
256     </mx:Script>
257     <mx:Model id="formParam">
258         <data>
259             <t60_sectionid>{Data.t60_sectionid}</t60_sectionid>
260             <t50_articleid>{Data2.t50_articleid}</t50_articleid>
261             <t50_xml_typeid>{t50_xml_typeid.selectedItem.t140_xmlid}</
                t50_xml_typeid>
262             <t51_versionid>{Data.t51_versionid}</t51_versionid>
263             <t50_title>{t50_title.data}</t50_title>
264             <t51_typeid>{t51_typeid.selectedItem.ikey}</t51_typeid>
265             <t51_excerpt>{t51_excerpt.data}</t51_excerpt>
266             <t51_status>{t51_status.data}</t51_status>
267             <t51_app_status>{}</t51_app_status>
```

```
268             <t51_app_by>{t51_app_by.selectedItem.t10_userid}</t51_app_by>
269             <t51_secure>{t51_secure.data}</t51_secure>
270             <t51_searchable>{t51_searchable.data}</t51_searchable>
271             <t51_content>{t51_content.data}</t51_content>
272             <t51_dt_pub>{t51_dt_pub.fieldData}</t51_dt_pub>
273             <t51_dt_exp>{t51_dt_exp.fieldData}</t51_dt_exp>
274             <t51_xml>{t51_xml.fieldData}</t51_xml>
275         </data>
276     </mx:Model>
277     <mx:Canvas height="100%" width="100%" id="formContainer">
278         <mx:VBox left="0" right="20" height="100%">
279             <ns:_form id="viewForm" width="100%">
280                 <ns:_form_textInput label="Title" id="t50_title" data=
                    "{Data2.t50_title}" required="true"/>
281                 <ns:_form_datetime label="Article Created" id="t50_dt" data=
                    "{Data2.t50_dt}" currentState="read_only"/>
282                 <ns:_form_item label="Type" id="t50_xml_typeid_form_item">
283                     <mx:ComboBox id="t50_xml_typeid" labelField="t140_name"
                        styleName="system"/>
284                 </ns:_form_item>
285                 <mx:Spacer height="3"/>
286                 <mx:HRule width="100%"/>
287             </ns:_form>
288             <ns:_form id="viewForm2" width="100%">
289                 <ns:_form_text label="Version" id="t51_versioncode" data=
                    "{Data.t51_versioncode}"/>
290                 <ns:_form_textInput label="Excerpt" id="t51_excerpt" data=
                    "{Data.t51_excerpt}"/>
291                 <ns:_form_textInput label="t51_content" id="t51_content" data=
                    "{Data.t51_content}" visible="false" includeInLayout="false"/>
292                 <ns:_form_datetime label="Publish Date &amp;amp; Time" id="t51_dt_pub"
                    data="{Data.t51_dt_pub}" change="_app.dateCheck(t51_dt_pub,
                    t51_dt_exp);"/><!--minDate="{datePubNow}"-->
293                 <ns:_form_datetime label="Expire Date &amp;amp; Time" id="t51_dt_exp"
                    data="{Data.t51_dt_exp}" change="_app.dateCheck(t51_dt_pub,
                    t51_dt_exp);"/>
294                 <ns:_form_item id="_form_item2">
295                     <ns:_form_checkBox_container>
296                         <ns:_form_checkBox id="t51_secure" label="" fieldLabel=
                            "Secured" data="{Data.t51_secure}" />
297                         <ns:_form_checkBox id="t51_status" label="" fieldLabel=
                            "Active" data="{Data.t51_status}" />
298                         <ns:_form_checkBox id="t51_searchable" label="" fieldLabel=
                            "Searchable" data="{Data.t51_searchable}" />
299                         <!--<ns:_form_checkBox id="t51_app_status" label=""
                            fieldLabel="Approved" data="{Data.t51_app_status}" enabled=
                            "false" visible="false" includeInLayout="false"/>-->
300                     </ns:_form_checkBox_container>
301                 </ns:_form_item>
302                 <ns:_form_item label="Content Editor" direction="horizontal"
                    horizontalGap="0">
303                     <mx:ComboBox id="t51_typeid" labelField="idesc">
304                         <mx:ArrayCollection id="t51_typeid_arr">
305                             <mx:Object idesc="WYSIWYG editor" ikey="1"/>
306                             <mx:Object idesc="Plain Text editor" ikey="2"/>
307                             <mx:Object idesc="Rich Text editor (HTML)" ikey="3"/>
```

```
308                             <mx:Object idesc="Rich Text editor (Flash)" ikey="4"/>
309                             <mx:Object idesc="PHP editor" ikey="5"/>
310                         </mx:ArrayCollection>
311                     </mx:ComboBox>
312                     <ns:_ui_btn id="editContentBtn" currentState="editContentBtn"
                        click="_app.loadEditor(t51_typeid.selectedItem.ikey);" />
313                 </ns:_form_item>
314                 <ns:_form_item label="Approver" id="approverSelect" direction=
                    "horizontal" horizontalGap="0" toolTip="Approver will not affect
                    approved version.">
315                     <mx:ComboBox id="t51_app_by" labelField="t10_ulogin" styleName=
                        "system"/>
316                 </ns:_form_item>
317                 <ns:_form_text label="Approval Status" id="approverDisplay" data=""
                    />
318                 <ns:_form_datetime label="Version Created" id="t51_dt_created" data
                    ="{Data.t51_dt_created}" currentState="read_only"/>
319                 <ns:_form_datetime label="Last Updated" id="t51_dt_updated" data=
                    "{Data.t51_dt_updated}" currentState="read_only"/>
320                 <ns:_form_text label="Created by" id="t51_owner_id" data=
                    "{Data.t10_ulogin}" style="system"/>
321                 <mx:Spacer height="3"/>
322                 <mx:HRule width="100%"/>
323             </ns:_form>
324             <ns:_form id="viewForm3" width="100%">
325                 <ns:_form_udx id="t51_xml" data=
                    "{XML(t50_xml_typeid.selectedItem.structure).children()}"
                    extendedData="{Data.t51_xml}" xmlEditor="false"/>
326             </ns:_form>
327         </mx:VBox>
328     </mx:Canvas>
329     <mx:states>
330         <mx:State name="view">
331             <mx:AddChild position="lastChild">
332                 <ns:_ui_btn_bar id="viewBtnBar">
333                     <ns:_ui_btn id="selectBtn" currentState="selectBtn" click=
                        "url('select');" />
334                     <!--<ns:_ui_btn id="previewBtn" currentState="webpageBtn" click
                        ="url('view')"/>-->
335                     <mx:Spacer width="100%"/>
336                     <!--<ns:_ui_btn id="editHTMLBtn" currentState="editHTMLBtn"
                        click="_app.main.loadEditor();" />-->
337                     <ns:_ui_btn id="newBtn" currentState="newBtn" click=
                        "newForm(); " visible="false" includeInLayout="false"/>
338                     <ns:_ui_btn id="saveBtn" currentState="saveBtn" click=
                        "updateSE();" toolTip="Save Current Version"/>
339                     <ns:_ui_btn id="saveAsBtn" currentState="saveAsBtn" click=
                        "updateSE(true);" toolTip="Save As New Version"/>
340                     <ns:_ui_btn id="removeBtn" currentState="removeBtn" click=
                        "remove();" />
341                     <ns:_ui_btn id="resetBtn" currentState="resetBtn" click=
                        "refreshForm();" />
342                 </ns:_ui_btn_bar>
343             </mx:AddChild>
344         </mx:State>
345         <mx:State name="new">
```

```
346             <mx:RemoveChild target="{t51_dt_updated}"/>
347             <mx:RemoveChild target="{t51_dt_created}"/>
348             <mx:SetProperty target="{t51_dt_pub}" name="data" value="{datePubNow}"/>
349             <mx:SetProperty target="{t51_dt_exp}" name="data" value="{dateExpNow}"/>
350             <mx:SetProperty target="{t51_status}" name="data" value="1"/>
351             <mx:SetProperty target="{t51_secure}" name="data" value="0"/>
352             <!--<mx:SetProperty target="{t51_app_status}" name="data" value="0"/>-->
353             <mx:SetProperty target="{t50_xml_typeid}" name="selectedIndex" value=
                "0"/>
354             <mx:RemoveChild target="{t50_dt}"/>
355             <mx:RemoveChild target="{t51_owner_id}"/>
356             <mx:RemoveChild target="{t51_versioncode}"/>
357             <mx:RemoveChild target="{approverDisplay}"/>
358             <mx:AddChild position="lastChild">
359                 <ns:_ui_btn_bar>
360                     <!--<ns:_ui_btn currentState="editHTMLBtn" click=
                        "_app.main.loadEditor();" />-->
361                     <ns:_ui_btn currentState="saveBtn" click="add();" enabled=
                        "{!itemExistFlag}" />
362                     <ns:_ui_btn currentState="cancelBtn" click="refreshForm();" />
363                 </ns:_ui_btn_bar>
364             </mx:AddChild>
365         </mx:State>
366         <mx:State name="no_version" basedOn="view">
367             <!--<mx:SetProperty target="{editHTMLBtn}" name="enabled" value="false"
                />-->
368             <mx:RemoveChild target="{viewForm2}"/>
369             <mx:RemoveChild target="{viewForm3}"/>
370             <mx:RemoveChild target="{saveAsBtn}"/>
371         </mx:State>
372         <mx:State name="root_data" basedOn="view">
373             <mx:AddChild>
374                 <ns:_ui_text width="100%" height="100%" text="Please select an
                    article or a version." />
375             </mx:AddChild>
376             <mx:RemoveChild target="{formContainer}"/>
377             <mx:RemoveChild target="{viewBtnBar}"/>
378         </mx:State>
379         <mx:State name="no_data" basedOn="view">
380             <mx:AddChild position="firstChild">
381                 <ns:_ui_note currentState="no_article"/>
382             </mx:AddChild>
383             <mx:RemoveChild target="{formContainer}"/>
384             <!--<mx:SetProperty target="{editHTMLBtn}" name="enabled" value="false"
                />-->
385             <mx:RemoveChild target="{viewBtnBar}"/>
386         </mx:State>
387         <mx:State name="with_version" basedOn="view">
388             <mx:SetProperty target="{t50_xml_typeid}" name="enabled" value="false"/>
389             <mx:SetProperty target="{t51_dt_created}" name="data" value=
                "{Data.t51_dt_created}"/>
390         </mx:State>
391         <mx:State name="no_version_approval" basedOn="no_version">
392             <mx:AddChild position="lastChild">
393                 <ns:_ui_btn_bar>
394                     <ns:_ui_btn currentState="webpageBtn" click="url('view')"/>
```

```
395                     <mx:Spacer width="100%"/>
396                 </ns:_ui_btn_bar>
397             </mx:AddChild>
398             <mx:RemoveChild target="{viewBtnBar}"/>
399         </mx:State>
400         <mx:State name="with_version_approval" basedOn="with_version">
401             <mx:AddChild position="lastChild">
402                 <ns:_ui_btn_bar>
403                     <ns:_ui_btn currentState="webpageBtn" click="url('view')"/>
404                     <mx:Spacer width="100%"/>
405                     <ns:_ui_btn id="approveBtn" currentState="approveBtn" click=
                        "parentObject.approval(formParam, true);"/>
406                     <ns:_ui_btn id="rejectBtn" currentState="rejectBtn" click=
                        "parentObject.approval(formParam, false);"/>
407                 </ns:_ui_btn_bar>
408             </mx:AddChild>
409             <mx:RemoveChild target="{viewBtnBar}"/>
410         </mx:State>
411     </mx:states>
412 </mx:VBox>
```

```
<?xml version="1.0" encoding="utf-8"?>
<ns:_ui_vdividedbox dataChange="init()" xmlns:mx="http://www.adobe.com/2006/mxml" xmlns:ns="_com.*">
    <mx:Script>
        <![CDATA[
            import mx.rpc.events.ResultEvent;
            import mx.controls.Alert;

            [Bindable]public var parentObject:Object;
            [Bindable]private var chartXML:XMLList;
            private var chartHtTPService:_sys_httpService;
            private var HTTPService:_sys_httpService;
            //[Bindable]private var app:_app = new _app;
            private function init():void{
                ui();
                if(data == null){
                    sectionForm.data=null;
                    //childSectionList.dataProvider=null;
                }
                else if(data == "root"){
                    sectionForm.data="root";
                    if(_app.global["autoLoad"]){
                        getChart();
                    }
                    else{
                        chartXML = sampleXML.data;
                    }
                    //getChildSections();
                }
                else{
                    getSection();
                    if(_app.global["autoLoad"]){
                        getChart();
                    }
                    else{
                        chartXML = sampleXML.data;
                    }
                    //getChildSections();
                }
            }
            private function ui():void{
                if(_app.global["linkSelect"] == true || parent.toString().split("."
                )[1] == "moduleLoader"){
                    _app.uiVisibility(sectionInfoContainer, false);
                    _app.uiRemove(sectionInfoContainer);
                    sectionMainContainer.percentHeight = 100;
                }
            }
            public function chkSection(formParam:Object, resultHandler:Function):
            void{
                var HTTPparams:Object = new Object;
                HTTPparams.func="cSection.chk_section";
                HTTPparams.t40_code=formParam.t40_code;
                HTTPparams.t40_sectionid=formParam.t40_sectionid;

                HTTPService = new _sys_httpService(_app.servicesPath, null,
```

```
                HTTPparams, resultHandler);
            }
            private function getSection():void{
                var HTTPparams:Object = new Object;
                HTTPparams.func="cSection.get_section";
                HTTPparams.t40_sectionid=data;
                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, resultHandler);

                function resultHandler(event:ResultEvent):void{
                    if(!_app.resultStatus(event)){return;}

                    var newArr:Array=_app.getContentData(event);

                    sectionForm.data = newArr[0];
                    //sectionChart.data=newArr[0];
                }
            }
            public function addSection(formParam:Object):void{
                //Alert.show(formParam.t40_secure);
                var HTTPparams:Object = new Object;
                HTTPparams = formParam;
                HTTPparams.func="cSection.create_section";
                HTTPparams.t40_parentid=data;
                if(_app.global["ui_mode"] == "fe" && !_app.global[
                "unlockSectionLevel"]){
                    HTTPparams.t40_parentid="root";
                }
                HTTPparams.t40_dt_pub=_app.encodeDate(formParam.t40_dt_pub);
                HTTPparams.t40_dt_exp=_app.encodeDate(formParam.t40_dt_exp);
                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, sectionResultHandler, null, null, true, true);
            }
            public function setSection(formParam:Object):void{
                var HTTPparams:Object = new Object;
                HTTPparams = formParam;
                HTTPparams.func="cSection.update_section";
                HTTPparams.t40_dt_pub=_app.encodeDate(formParam.t40_dt_pub);
                HTTPparams.t40_dt_exp=_app.encodeDate(formParam.t40_dt_exp);
                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, sectionResultHandler, null, null, true, true);
            }
            public function removeSection(formParam:Object):void{
                var HTTPparams:Object = new Object;
                HTTPparams.func="cSection.delete_section";
                HTTPparams.t40_sectionid=formParam.t40_sectionid;
                HTTPService = new _sys_httpService(_app.servicesPath, null,
                HTTPparams, sectionResultHandler, null, null, true, true);
            }
            private function sectionResultHandler(event:ResultEvent):void{
                //Alert.show(event.result.toString());
                    if(!_app.resultStatus(event)){return;}
    //              init();
                    parentObject.init();
            }
            public function getChildSections():void{
```

```
104                 var HTTPparams:Object = new Object;
105                 HTTPparams.func="cSection.get_child_sections";
106                 HTTPparams.t40_parentid=data;
107                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
108
109                 function resultHandler(event:ResultEvent):void{
110                     if(!_app.resultStatus(event)){return;}
111
112                     var newArr:Array=_app.getContentData(event);
113
114                     //childSectionList.dataProvider = newArr;
115                 }
116             }
117             public function cancelChart():void{
118                 try{
119                     chartHtTPService.disconnect();
120                 }catch(error:Error){}
121             }
122             public function getChart():void{
123                 chart.intervalLabel = chartType.chart.(@data==chartController.
                    chartCat.selectedItem.@data).group.(@data==chartController.group.
                    selectedItem.data).@interval;
124
125                 var HTTPparams:Object = new Object;
126                 HTTPparams.func= chartType.chart.(@data==chartController.chartCat.
                    selectedItem.@data).group.(@data==chartController.group.
                    selectedItem.data).@func;
127                 HTTPparams.target=data;
128                 HTTPparams.targetField = "t90_sectionid";
129                 HTTPparams.day = chartController.day.selectedItem.data;
130                 HTTPparams.month = chartController.month.selectedItem.data;
131                 HTTPparams.year = chartController.year.selectedItem.data;
132                 chartHtTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
133
134                 function resultHandler(event:ResultEvent):void{
135                     //Alert.show(XML(event.result).toXMLString());
136                     if(!_app.resultStatus(event)){return;}
137                     chartXML = XML(event.result).content;
138                 }
139             }
140         ]]>
141     </mx:Script>
142     <mx:XML id="sampleXML" xmlns="">
143         <root>
144             <data ref="1">
145                 <sc>0</sc>
146             </data>
147         </root>
148     </mx:XML>
149     <mx:XML id="chartType" xmlns="">
150         <root>
151             <chart data="totalPageViews" label="Total Page Views">
152                 <group data="byHour" func="get_target_view_hour" interval="Hour"/>
153                 <group data="byDay" func="get_target_view_day" interval="Day"/>
```

```
154                 <group data="byMonth" func="get_target_view_month" interval="Month"
                    />
155             </chart>
156             <chart data="uniqueVisitors" label="Unique Visitors">
157                 <group data="byHour" func="get_target_view_hour_distinct" interval=
                    "Hour"/>
158                 <group data="byDay" func="get_target_view_day_distinct" interval=
                    "Day"/>
159                 <group data="byMonth" func="get_target_view_month_distinct"
                    interval="Month"/>
160             </chart>
161         </root>
162     </mx:XML>
163     <ns:_ui_panel_container id="sectionMainContainer" minHeight="255" height="255">
164         <ns:_ui_panel title="Section Detail">
165             <ns:pub_sec_form id="sectionForm" parentObject="{this}"/>
166         </ns:_ui_panel>
167     </ns:_ui_panel_container>
168     <ns:_ui_vdividedbox id="sectionInfoContainer">
169         <ns:_ui_panel_container>
170             <ns:_ui_panel title="Section Web Statistic">
171                 <mx:HBox width="100%">
172                     <ns:_ui_chart_controller id="chartController" parentObject=
                        "{this}" chartType="{chartType}"/>
173                 </mx:HBox>
174                 <ns:_ui_chart id="chart" data="{chartXML.data}" targetData="data"
                    targetLabel="{parentObject.sectionList.selectedItem.@label}"
                    currentState="genLine" parentObject="{this}"/>
175             </ns:_ui_panel>
176         </ns:_ui_panel_container>
177         <!--
178         <mx:Canvas width="100%" height="50%">
179             <ns:_ui_panel title="Child Sections">
180                 <ns:_ui_list id="childSectionList" setState="section" parentObject=
                    "{this}" columnCount="1" height="100%" width="100%" dragEnabled=
                    "true" dropEnabled="true" dragMoveEnabled="true">
181                 </ns:_ui_list>
182             </ns:_ui_panel>
183         </mx:Canvas>
184         -->
185     </ns:_ui_vdividedbox>
186 </ns:_ui_vdividedbox>
187
```

```
<?xml version="1.0" encoding="utf-8"?>
<mx:VBox creationComplete="preinit();init()" dataChange="init()" xmlns:mx=
"http://www.adobe.com/2006/mxml" xmlns:ns="_com.*" width="100%" height="100%">
    <mx:Script>
        <![CDATA[
            import mx.events.ListEvent;
            import mx.events.CloseEvent;
            import mx.events.FlexEvent;
            import mx.rpc.events.ResultEvent;
            import mx.controls.Alert;

            private var HTTPService:_sys_httpService;

            public var parentObject:Object;
            [Bindable]private var Data:Object;
            [Bindable]private var itemExistFlag:Boolean=false;
            [Bindable]private var datePubNow:Date;
            [Bindable]private var dateExpNow:Date;
            //[Bindable]private var app:_app = new _app;

            private function preinit():void{
                getDynamicTypes();
            }
            private function init():void{
                Data=null;Data=data;
                ui();

                datePubNow = new Date(_app.systemDate);
                dateExpNow = new Date(_app.systemDate);
                dateExpNow.setFullYear(dateExpNow.getFullYear()+5);

                if(data == null){
                    this.currentState="no_data";
                }else{
                    loadInfo();
                    this.currentState="view";
                    if(Data=="root"){
                        this.currentState="root_data";
                        t40_code.errorString="";
                        itemExistFlag=false;
                    }else{
                        check();
                    }
                }
            }
            private function ui():void{
                if(_app.global["ui_mode"] == "fe"){
                    if(!_app.global["unlockUDX"]){
                        _app.uiVisibility(viewForm2, false);
                    }
                    _app.uiVisibility(t40_secure, false);
                }
                if(_app.global["linkSelect"] == true || parent.toString().split("."
                )[1] == "moduleLoader"){
                    _app.uiVisibility(selectBtn, true);
                    //_app.uiVisibility(previewBtn, false);
```

```
55                  }else{
56                      _app.uiVisibility(selectBtn, false);
57                      //_app.uiVisibility(previewBtn, true);
58                  }
59
60
61                  try{
62                      _app.uiVisibility(parentObject.parentObject.newBtn, true);
63                      _app.uiVisibility(saveBtn, true);
64                      _app.uiVisibility(removeBtn, true);
65
66                      if(Data.t41_add != "1"){
67                          _app.uiVisibility(parentObject.parentObject.newBtn, false);
68                      }
69                      if(Data.t41_edit != "1"){
70                          _app.uiVisibility(saveBtn, false);
71                      }
72                      if(Data.t41_delete != "1"){
73                          _app.uiVisibility(removeBtn, false);
74                      }
75                  }catch(error:Error){}
76              }
77              private function check():void{
78                  t40_code.field.errorString="";
79                  itemExistFlag=true;
80
81                  parentObject.chkSection(formParam, resultHandler);
82                  function resultHandler(event:ResultEvent):void{
83                      //Alert.show(event.result.toString());
84                      if(_app.resultStatus(event,"none")){
85                          t40_code.field.errorString="Code already exist.";
86
87                          itemExistFlag=true;
88                      }
89                      else{
90                          if(currentState != "protected_data"){
91                              itemExistFlag=false;
92                          }
93                          if(!validation()){return}
94                      }
95                  }
96              }
97              private function getDynamicTypes():void{
98                  var HTTPparams:Object = new Object;
99                  HTTPparams.func="cSection.get_xml_type";
100                 HTTPService = new _sys_httpService(_app.servicesPath, null,
                    HTTPparams, resultHandler);
101
102                 function resultHandler(event:ResultEvent):void{
103                     if(!_app.resultStatus(event)){return;}
104
105                     var newArr:Array=_app.getContentData(event);
106                     t40_xml_typeid.dataProvider = _app.getNAObject(newArr,
                        "t140_xmlid", "t140_name");
107                 }
108             }
```

```
109             private function loadInfo():void{
110                 try{
111                     t40_xml_typeid.selectedIndex = 0;
112                     var i:Number = 0;
113                     for each(var cat:Object in t40_xml_typeid.dataProvider){
114                         if(cat.t140_xmlid ==  Data.t40_xml_typeid){
115                             t40_xml_typeid.selectedIndex = i;
116                         }
117                         i++;
118                     }
119                     t40_xml.extendedDataChange();
120                 }catch(error:Error){}
121             }
122             public function refreshForm():void{
123                 this.dispatchEvent(new FlexEvent(FlexEvent.DATA_CHANGE));
124             }
125             private function newForm():void{
126                 //parentObject.parentObject.sectionList.selectedIndex = -1;
127                 //parentObject.parentObject.sectionList.dispatchEvent(new
                    ListEvent(ListEvent.CHANGE));
128                 currentState='new';
129                 Data=null;
130                 t40_status.data = 1;
131                 t40_dt_pub.data=datePubNow;
132                 t40_dt_exp.data=dateExpNow;
133             }
134             private function add():void{
135                 if(!validation()){return}
136 
137                 parentObject.addSection(formParam);
138                 refreshForm();
139             }
140             private function update():void{
141                 if(!validation()){return}
142 
143                 parentObject.setSection(formParam);
144                 refreshForm();
145             }
146             private function remove():void{
147                 _app.confirm("Are you sure you want to remove section?",
                    confirmHandler);
148 
149                 function confirmHandler(event:CloseEvent):void{
150                     if (event.detail == Alert.YES) {
151                         parentObject.removeSection(formParam);
152                         refreshForm();
153                     }
154                 }
155             }
156             private function validation():Boolean{
157                 return (_app.validation(viewForm) && t40_xml.validation());
158             }
159             private function url(actionIn:String):void{
160                 var url:String = _app.generateURL(Data);
161 
162                 switch(actionIn){
```

```
163                     case "select":
164                         _app.SelectFile(url);
165                     break;
166                     case "view":
167                         _app.loadPreview(url);
168                     break;
169                 }
170             }
171         ]]>
172     </mx:Script>
173     <mx:Model id="formParam">
174         <data>
175             <t40_sectionid>{Data.t40_sectionid}</t40_sectionid>
176             <t40_name>{t40_name.data}</t40_name>
177             <t40_code>{t40_code.data}</t40_code>
178             <t40_status>{t40_status.data}</t40_status>
179             <t40_secure>{t40_secure.data}</t40_secure>
180             <t40_hidden>{t40_hidden.data}</t40_hidden>
181             <t40_home>{t40_home.data}</t40_home>
182             <t40_dt_pub>{t40_dt_pub.fieldData}</t40_dt_pub>
183             <t40_dt_exp>{t40_dt_exp.fieldData}</t40_dt_exp>
184             <t40_xml>{t40_xml.fieldData}</t40_xml>
185             <t40_xml_typeid>{t40_xml_typeid.selectedItem.t140_xmlid}</
                t40_xml_typeid>
186         </data>
187     </mx:Model>
188     <mx:Canvas height="100%" width="100%" id="formContainer">
189         <mx:HBox left="0" right="20" height="100%">
190             <ns:_form id="viewForm" width="100%">
191                 <ns:_form_textInput id="t40_name" label="Name" data=
                    "{Data.t40_name}" required="true"/>
192                 <ns:_form_textInput id="t40_code" label="Code" data=
                    "{Data.t40_code}" required="true" change="check();" wordOnly="true"
                    />
193                 <ns:_form_datetime label="Publish Date &amp;amp; Time" id="t40_dt_pub"
                    data="{Data.t40_dt_pub}" change="_app.dateCheck(t40_dt_pub,
                    t40_dt_exp);"/><!--minDate="{datePubNow}"-->
194                 <ns:_form_datetime label="Expire Date &amp;amp; Time" id="t40_dt_exp"
                    data="{Data.t40_dt_exp}" change="_app.dateCheck(t40_dt_pub,
                    t40_dt_exp);"/>
195                 <ns:_form_item>
196                     <ns:_form_checkBox_container>
197                         <ns:_form_checkBox id="t40_secure" label="" fieldLabel=
                            "Secured" data="{Data.t40_secure}" />
198                         <ns:_form_checkBox id="t40_status" label="" fieldLabel=
                            "Active" data="{Data.t40_status}" />
199                         <ns:_form_checkBox id="t40_hidden" label="" fieldLabel=
                            "Hidden" data="{Data.t40_hidden}" />
200                         <ns:_form_checkBox id="t40_home" label="" fieldLabel="Home
                            Page" data="{Data.t40_home}"/>
201                     </ns:_form_checkBox_container>
202                 </ns:_form_item>
203                 <ns:_form_datetime label="Created" id="t40_dt_created" data=
                    "{Data.t40_dt_created}" currentState="read_only"/>
204                 <ns:_form_datetime label="Last Updated" id="t40_dt_updated" data=
                    "{Data.t40_dt_updated}" currentState="read_only"/>
```

```
205             </ns:_form>
206             <!--<mx:VRule height="100%"/>-->
207             <ns:_form id="viewForm2" width="100%">
208                 <mx:VBox width="100%" height="100%">
209                     <ns:_form_item label="Type" id="_form_item1">
210                         <mx:ComboBox id="t40_xml_typeid" labelField="t140_name"
                            styleName="system"/>
211                     </ns:_form_item>
212                     <ns:_form_udx id="t40_xml" data=
                        "{XML(t40_xml_typeid.selectedItem.structure).children()}"
                        extendedData="{Data.t40_xml}" height="{viewForm.height -
                        _form_item1.height}" xmlEditor="false"/>
213                 </mx:VBox>
214             </ns:_form>
215         </mx:HBox>
216     </mx:Canvas>
217     <mx:states>
218         <mx:State name="view">
219             <mx:AddChild position="lastChild">
220                 <ns:_ui_btn_bar id="viewBtnBar">
221                     <ns:_ui_btn id="selectBtn" currentState="selectBtn" click=
                        "url('select');"/>
222                     <ns:_ui_btn id="previewBtn" currentState="webpageBtn" click=
                        "url('view');"/>
223                     <mx:Spacer width="100%"/>
224                     <ns:_ui_btn id="newBtn" currentState="newBtn" click=
                        "newForm(); " visible="false" includeInLayout="false"/>
225                     <ns:_ui_btn id="saveBtn" currentState="saveBtn" click=
                        "update();" enabled="{!itemExistFlag}"/>
226                     <ns:_ui_btn id="removeBtn" currentState="removeBtn" click=
                        "remove();"/>
227                     <ns:_ui_btn id="resetBtn" currentState="resetBtn" click=
                        "refreshForm();" />
228                 </ns:_ui_btn_bar>
229             </mx:AddChild>
230             <mx:SetProperty target="{t40_code.field}" name="errorString" value=""/>
231         </mx:State>
232         <mx:State name="new">
233             <mx:SetProperty target="{t40_xml_typeid}" name="selectedIndex" value=
                "0"/>
234             <mx:SetProperty target="{t40_dt_pub}" name="data" value="{datePubNow}"/>
235             <mx:SetProperty target="{t40_dt_exp}" name="data" value="{dateExpNow}"/>
236             <mx:RemoveChild target="{t40_dt_updated}"/>
237             <mx:RemoveChild target="{t40_dt_created}"/>
238             <mx:AddChild position="lastChild">
239                 <ns:_ui_btn_bar>
240                     <ns:_ui_btn currentState="saveBtn" click="add();" enabled=
                        "{!itemExistFlag}" />
241                     <ns:_ui_btn currentState="cancelBtn" click="refreshForm();" />
242                 </ns:_ui_btn_bar>
243             </mx:AddChild>
244             <mx:SetProperty target="{t40_status}" name="data" value="1"/>
245             <mx:SetProperty target="{t40_secure}" name="data" value="0"/>
246             <mx:SetProperty target="{t40_hidden}" name="data" value="0"/>
247             <mx:SetProperty target="{t40_home}" name="data" value="0"/>
248             <mx:SetProperty target="{t40_code.field}" name="errorString" value=""/>
```

```
249             </mx:State>
250             <mx:State name="no_data" basedOn="view">
251                 <mx:AddChild>
252                     <ns:_ui_note currentState="no_section"/>
253                 </mx:AddChild>
254                 <mx:RemoveChild target="{formContainer}"/>
255                 <mx:RemoveChild target="{viewBtnBar}"/>
256             </mx:State>
257             <mx:State name="root_data" basedOn="view">
258                 <mx:AddChild>
259                     <ns:_ui_note currentState="no_section"/>
260                 </mx:AddChild>
261                 <mx:RemoveChild target="{formContainer}"/>
262                 <mx:RemoveChild target="{viewBtnBar}"/>
263             </mx:State>
264         </mx:states>
265     </mx:VBox>
```