Combining MDE and UML To Reverse Engineer Web-Based Legacy Systems

Jianjun Pu[1], Hongji Yang[1], Baowen Xu[2], Lei Xu[2] and William Cheng-Chung Chu[3]
1, Software Technology Research Laboratory, De Montfort University, England
2, Computing School, Southeast University, China
3, Dept. of Computer Science and Information Engineering, Tunghai University, Taiwan

Abstract

The research in this paper focuses on an approach to reverse engineering web-based legacy systems with the integration of model-driven engineering and UML. Three types of link-based models of web-based legacy systems are presented. Web-based legacy systems are parsed to find judgement conditions of model, and UML diagrams are described based on the modelling rules.

1. Introduction

Because of the numerous users of Internet, web is fundamentally important to the information exchange and data presentation [5]. It is necessary to analyse the useful contents of web legacy system in order to maintain and reuse them [2]. In this paper, three link-based models of web-based legacy systems are presented [3, 4]. In link-based model, one web page is regarded as web model entity. There are two types of ordinal and loop linking relationships between web model entities. Based on searching link-based model of web-based legacy systems, eleven rules are presented to realise selected four UML diagrams [1, 5]. The major contribution of this paper is the definition of link-based models of web-based legacy systems on the characteristics and operations of domain-specific legacy systems and reverse engineering them with the integration of MDE and UML.

2. Grouping web code operations

The statements of in Web code are composed of five groups: text, image, link, frame, and table. Each group contains different operations.

Group One is the description of the text:

- Interpretative operations, which are used to interpret the structure of the program PP. They include <H>, <META>, <APPLET>, <SCRIPT>, <NOSCRIPT>, <P>,
, <BLOCKQUOTE>, <!-->, <PRE>, <SPAN>, <DIV>, <STYLE> statements.
- Word Style, which are the description of display of the letter or word. They include <B>, <BIG>, <SMALL>, <EM>, <I>, <STRONG>, <BASEFONT>, <FONT>, <SUB>, <SUP>, <TT>, <U> statements.
- Line operations, which insert and delete the lines in the text. They include <DEL>, <HR>, <INS>, <STRIKE> statements.
- Special operations, which are the description of special display of text. They include <CODE>, <MARQUEE>, <Q>, <SAMP>, <KBD>, <CENTER> operations.

Group Two is the description of the image:

- Image presentation, which is used to present the images in WEB code. They include <IMG>, <MAP>, <EMBED>, <NOEMBED>, <AREA> operations.
- Image location, which is used to locate the image in the web page. They include <LAYER>, <SPACER> operations.

Group Three is the description of the link operations:

- Link creation, which is used to create the link and the link window. They include <A>, <BASE> operations.
- External link, which is used to set up the external link. It is the operation <LINK>.

Group Four is the description of frame operation:

- Frame creation, which creates the frames or the frame set in the WEB code. They are <FRAME>, <FRAMESET> operations.
- Embedded frame creation, which is used to create the embedded frames. It is <IFRAME>.

Group Five is the description of table operations:

- Common table operations, which are used to create and format the common table. They include <TABLE>, <TR>, <TH>, <TD>, <THEAD>, <TBODY>, <TFOOT>, <COL>, <COLGROUP>, <CAPTION> operations.
- Form table operations, which are used to present and format the form table. They include <FORM>, <INPUT>, <BUTTON>, <FIELDSET>, <LEGEND>, <SELECT>, <OPTION>, <OPTGROUP>, <LABEL>, <TEXTAREA> operations.
- Sequential table operations, which is used to represent sequential table. It is <OL>.
- Unsequential table operations, which is use to describe the unsequential table. It is <UL>.
- Table definition and creation operations, which is used to create table. They are <DL>, <DT>, <DD>, <LI> operations.

3. Parsing web code

Web-based legacy system has its own models to

0730-3157/08 $25.00 © 2008 IEEE
DOI 10.1109/COMPSAC.2008.63

show their structures and operation processes respectively. Normally parsing Web-based legacy system has two levels of grammar analysis: lexical and syntactic (Figure 1).

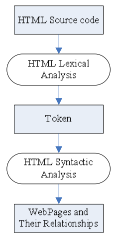

Figure 1: Process of Parsing WEB Code

The first step is to input web source code and acquire web tokens through lexical analysis. From the mathematic point of view, web source code is the set of data with the format of tables, images, etc. It is relocated in the coordinate system of which X-axis is the format of web and Y-axis is the contents of web.

The second step is syntactic analysis in which these web tokens are processed to build a data structure such as parser tree or abstract syntax trees. It outputs the web pages and their linking or being linked relationships.

4. Definition of link-based model

Link-based model is one graph to describe the importing or imported relationships of those webs in program P, indicated LBM( ). It is composed of nodes and lines. Every node represents one web page, and the line represents the link relationship. The web that the first node represents imports the webs that the next nodes represent. One example of link-based model is presented in Figure 2.

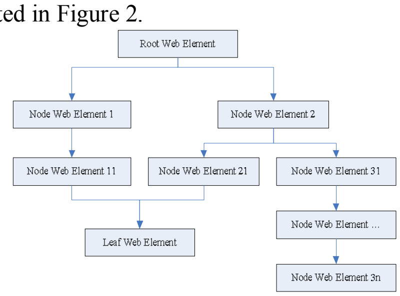

Figure 2: Link-Based Model

5. Three kinds of web link elements

The web link is the import of the new web pages written in HTML programming language. The starting point in analysing the structure of web-based legacy system is to develop a link-based model. Examining the link structure of web-based legacy system can be used to identify web threads with minimal dependencies that could be migrated easily. Three different kinds of web link layers are distinguished: root web element that leads to other web element links but are not quoted by any other, or is the home web page or the first web page although it is linked by other web page; leaf web elements that are quoted by other web elements but do not cause any other webs, or are the last web pages although they quote other web pages; node web elements that both lead to and are quoted by other web elements except root web element and leaf web elements.

6. Web layer

The web layer is one number that represents the depth of one web importing other webs, indicated WL( ). The web layer of root web element is 0, the web layer of the node web elements that are only imported by that root web element is 1, the web layer of web elements that are only imported by those node web elements the maximum of whose web layers is 1 is 2, etc (Figure 3).

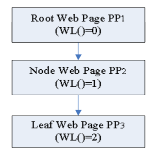

Figure 3: One Graph Example of Web Layer

7. Web relationships

In link-based model, one web page is regarded as the essential unit of WEB legacy system, and it is called as WEB model entity.

There are two kinds of the linking relationships between web model entities: ordinal and loop. The web relationship is defined as the linking or being linked relationship between the two web pages in web-based legacy system. It has two kinds: ordinal and loop.

In link-based model LBM( ), PP_t is called the dominator of PP_s if there exists one path in LBM( ) from PP_s to PP_t, indicated $PP_t \in \{DOM(PP_s)\}$. PP_t is called the immediate dominator of PP_s if PP_t is the immediate successor of PP_s, indicated PP_t=IM(PP_s).

Let PP_i, PP_j, PP_k be three web pages in web-based legacy system. If

$\forall k \neq i,\ PP_k \in \{DOM(PP_i)\} \Rightarrow (PP_i\,!\!\in \{DOM(PP_k)\}$

then the relationship between PP_i and PP_j is called as ordinal (Figure 4, 5, 6 and 7).

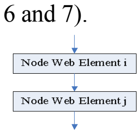

Figure 4: First Example of Ordinal Relationship

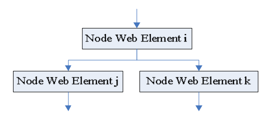

Figure 5: Second Example of Ordinal Relationship

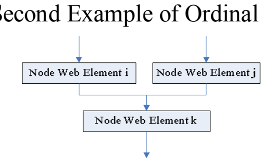

Figure 6: Third Example of Ordinal Relationship

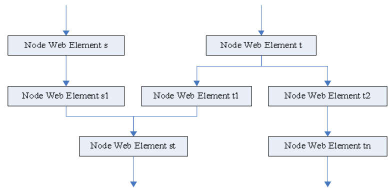

Figure 7: Fourth Example of Ordinal Relationship

If

$$(PP_t=IM(PP_s)) \text{ AND } (PP_s=IM(PP_t))$$

then the relationship between PP_i and PP_j is called as loop (Figure 8).

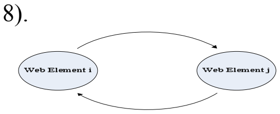

Figure 8: One Example of Loop Relationship

8. Classification of link-based model

Web Model 1: Sequential link-based model

Sequential link-based model is one link-based model in which the relationships between root web element and node web elements, node web elements, or node web elements and leaf web elements, are ordinal (Figure 9, 10, 11 and 12).

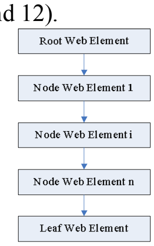

Figure 9: One Example of Sequential Link-Based Model

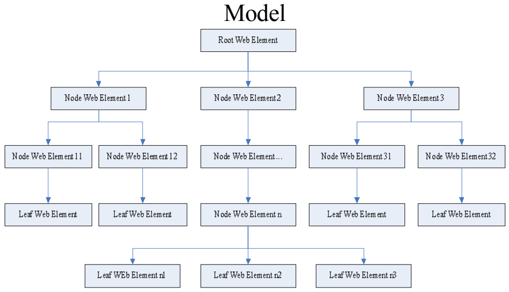

Figure 10: Second Example of Sequential Link-Based Model

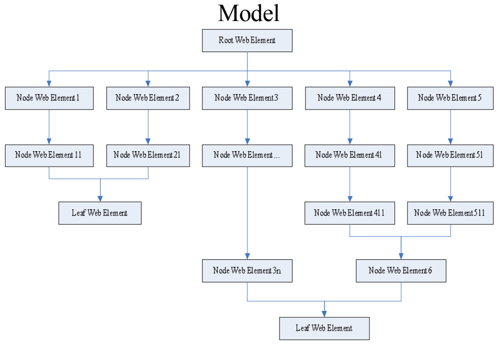

Figure 11: Third Example of Sequential Link-Based Model

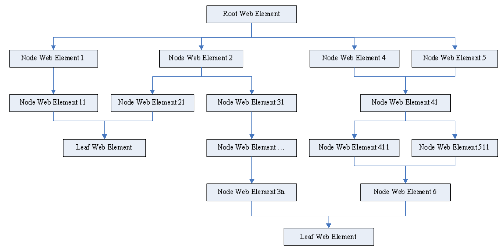

Figure 12: Fourth Example of Sequential Link-Based Model

Let PP_0 be root web element in web-based legacy system P, PP_i, PP_j, PP_k be three web elements, the set SAW{} be the set of all web elements in P (SAW-Set of All Webs),

$$SAW\{\}=\{ PP_i \mid PP_i \in P\},$$

the set SOW{} be the set of the web elements whose relationships are ordinal in P (SOW-Set of Ordinal Webs),

$$SOW\{\}=\{ PP_k \mid (\forall k\neq 0,\ PP_k\in \{DOM(PP_0)\})\Rightarrow(PP_0\,!\in \{DOM(PP_k)\})\}.$$

If

$$SOW\{\}= SAW\{\}-\{PP_0\},$$

then the link-based model of WEB legacy system is sequential.

Web Model 2: Cycle link-based model

Cycle link-based model is one link-based model in which one or more relationships between root web element and node web elements, node web elements, or node web elements and leaf web elements, are loop. There is no ordinal relationship in this link-based model (Figure 13 and 14).

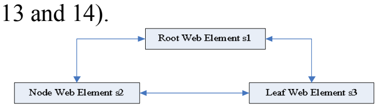

Figure 13: One Example of Cycle Link-Based Model

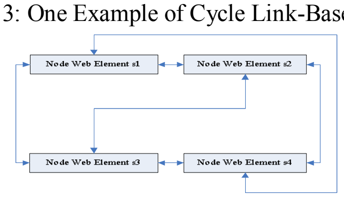

Figure 14: Another Example of Cycle Link-Based Model

Let the set SCW{} be the set of the web elements whose relationships are cycle in P (SCW-Set of Cycle Webs),

$$SCW\{\}=\{ PP_j \mid ((\forall k\neq j) \text{ AND } (PP_k\in SCW\{\})) \Rightarrow((PP_k\in \{IM(PP_j)\}) \text{ AND } (PP_j \in \{IM(PP_k)\}))\},$$

If

$$SCW\{\}= SAW\{\},$$

then the link-based model of WEB legacy system is cycle.

Web Model 3: Compositive link-based model

Compositive link-based model is one link-based model in which there exist two relationships of loop and ordinal at the same link-based model between root and node web elements, node web elements, or node web elements and leaf program elements (Figure 15, 16 and

17).

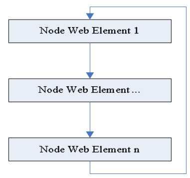

Figure 15: First Example of Compositive Link-Based Model

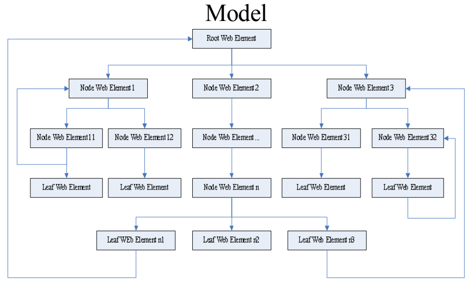

Figure 16: Second Example of Compositive Link-Based Model

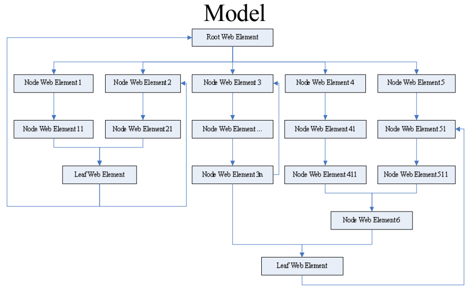

Figure 17: Third Example of Compositive Link-Based Model

Let the set SPW{} be the set of the web elements whose relationships contain ordinal and cycle at the same link-based model in P (SPW-Set of comPositive Webs),

$SPW\{\}=$

$\{PP_k|((PP_k \in \{DOM(PP_0)\}) \Rightarrow (PP_0 \ ! \in \{DOM(PP_k)\}))$ AND

$((\ \exists\ m \neq k) \Rightarrow ((PP_m \in \{IM(PP_k)\})AND(PP_k \in \{IM(PP_m)\})))\}$,

If

$$SPW\{\} \neq \phi,$$

then the link-based model of WEB legacy system is compositive.

9. Reverse engineering web-based legacy systems with UML

Rule 1: Layering class diagrams.

In WEB legacy system, one web page is regarded as the essential unit of the link-based model. The model entities are composed of the web systems. The structure of web legacy code is complicated and is displayed into different layers, which is represented with three different kinds of class diagrams.

The class diagram which presents the leaf web elements in source code is called as leaf class diagram, indicated LEAF-WEB-NAME.

The class diagram which presents the node web elements in source code is called as node class diagram, indicated NODE-WEB-NAME.

The class diagram which presents the root web element in source code is called as root class diagram, indicated ROOT-WEB-NAME.

One class diagram of a web code example is presented in Figure 18.

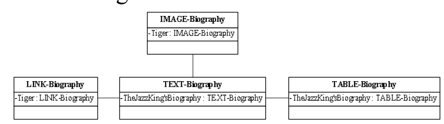

Figure 18: Class Diagram of A Web Code Example

Rule 2: Starting from root web element.

The analysis of web legacy code starts from root web element.

Leaf web elements do not contain the imported information. From leaf web elements, the link thread cannot be able to be discovered. Node web elements are even more difficult to find the clue to the root web element. Because they are imported by root web element and import leaf web elements, they share the difficulty of leaf web elements which are imported and increase the difficulty of root web element to track the link road. Root web element is the beginning of trailing the web links. No matter how many web pages are behind the root web element, it is the essential unit to analyse the whole web code. It is the best start point of tracking the link.

Because one leaf web element is one functional module and it is linked in the node web elements, it is defined as one class in analysing node web elements that call the leaf web element.

For one leaf web element P, leaf web class is the class with respect to that leaf web element in analysing the webs linking it, indicated CLASS LEAF-WEB-NAME.

The class which represents the node web element in source code is regarded as one class in node and root class diagrams which link that web, and it is called as node web class, indicated NODE-WEB-NAME.

In one node web element, its contained leaf and node web elements are presented with leaf and node class.

Rule 3: Partitioning one web page into several blocks.

The web page is composed of several parts, each one of which has the specific destination to describe the specific function, such as the title, the main description, the main data, the help, and the additional information. The web block is one part of one web page, which describes one specific function of that web page, indicated WB-NAME.

The web blocks of one web page contain former note, local header, local main, local footer, and additional note.

The former note is one web block in one web page that describes the top web, the main structure of the entire web system, the former web pages, the continuing web pages, etc. It is usually located at the upper web page. The local header is one web block in one web page that describes the main functions of the web page, the title of this web page, and the links to latter web pages. The local main is one web block in one web page that realises the main functions of the web page. The local footer is one web block in one web page that describes the explanation of the web realisation, the supplement of the web, the updating information, the contact styles, the information about the author, etc. It is usually located at the bottom of the web page. The additional note is one web block in one web page that provides the additional functions, such as helping, searching, statisticalising, recording, and advertising, and describes the links to latter web pages. It is usually located at the bottom or sides of the web page.

In some cases, one web block has several parts. For example, local main of one web page can be divided into local main part 1, local main part 2, …, and local main part n. In the root web element, there are several blocks, and each block has different data descriptions.

Rule 4: Presenting five operation classes.

The WEB has five groups operations: text, image, table, frame, and link. So the corresponding five kinds of classes are defined.

The text class of one web code PP is defined as the class that describes the text of the web element PP, indicated Text Class. Its attributes include font, location, content, period, subscript, superscript, etc. Its operations include "to delete line", "to format", "to insert underline", "to quote", "to scroll", "to make the same style", etc (Figure 19).

《Text Class》 Class-Name
Font: integer Colour: string Location: string Content: string
to delete line to insert underline to format to quote to scroll to make the same style

Figure 19: Text Class

The image class of one web code PP is defined as the class that describes the image of the web element PP, indicated Image Class. Its attributes include location, content, etc. Its operations include "to embed", "to substitute with", "to make the map", "to set spacer", "to create", etc (Figure 20).

《Image Class》 Class-Name
Location: string Content: string
to create to embed to substitute with to make the map to set spacer

Figure 20: Image Class

The table class of one web code PP is defined as the class that describes the table of the web element PP, indicated Table Class. Its attributes include location, table head, table body, table foot, caption, label, etc. Its operations include "to select", "to input", "to make button", "to make group", "to create text area", "to create volume", etc (Figure 21).

《Table Class》 Class-Name
Location: string Caption: string Table Head: string Table Body: string Table Foot: string Label: string
to select to input to make button to make group to create text area to create volume

Figure 21: Table Class

The frame class of one web code PP is defined as the class that describes the frame of the web element PP, indicated Frame Class. Its attributes include location, content, etc. Its operations include "to embed frame", "to create frame set", "to substitute with", etc (Figure 22).

《Frame Class》 Class-Name
Location: string Content: string
to embed frame to create frame set to substitute with

Figure 22: Frame Class

The link class of one web code PP is defined as the class that describes the link of the web element PP, indicated Link Class. It is the most important class among these classes because it leads to other web pages. Its attributes include location, base, content, layer, etc. Its operations include "to link", etc (Figure 23).

《Link Class》 Class-Name
Location: string Base: string Content: string Layer: integer
to link

Figure 23: Link Class

Rule 5: Realising class diagram of one web block.

Each block of one web page contains all or part of those five classes, and each block is defined as one class diagram. See Figure 24.

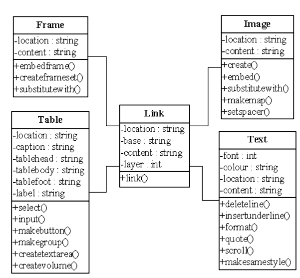

Figure 24: Class Diagram of One Web Block

Rule 6: Realising class diagram of one web element.

Assume that the web source code PW has the layer $j=n_0$, n_{js} represents the maximum number of the s^{th} web element of the j^{th} layer, n_{jst} represents the maximum number of t^{th} web block of the s^{th} web element of the j^{th} layer, PE_{jk} is the web element whose layer is j, $1 \leq s \leq n_{js,}$

whose web element is k, $1 \le k \le n_{js}$, and PP_{jkl} is the web block whose layer is j, $1 \le s \le n_{js,}$ whose web element is k, $1 \le k \le n_{js}$, $1 \le l \le n_{jst}$. The algorithm describing the class diagrams of web element of WEB code PW is:

For j:=0 to n_0 DO
  For k:=1 to n_{js} DO
    For l:=1 to n_{jkt} DO
      TextClass(PP_{jkl});
      ImageClass(PP_{jkl});
      TableClass(PP_{jkl});
      FrameClass(PP_{jkl});
      LinkClass(PP_{jkl});
      ClassDiagram(PP_{jkl});
    End-For
    ClassDiagram(PE_{jk});
  End-For
End-For
ClassDiagram(PW);

When j=0, it is the class diagram of root web element, and others are the ones of leaf or node web elements.

Rule 7: Presenting class diagrams of Web models.

Sequential link-based model is one link-based model in which the relationships between root web element and node web elements, node web elements, or node web elements and leaf web elements, are ordinal. The class diagram of sequential link-based model is described in Figure 25, 26, 27 and 28.

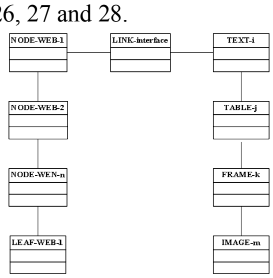

Figure 25: First Class Diagram of Sequential Link-Based Model

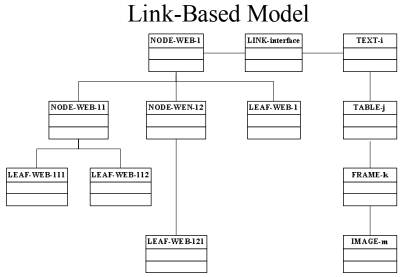

Figure 26: Second Class Diagram of Sequential Link-Based Model

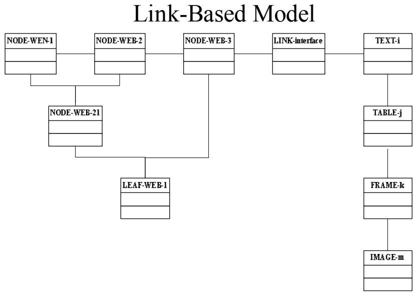

Figure 27: Third Class Diagram of Sequential Link-Based Model

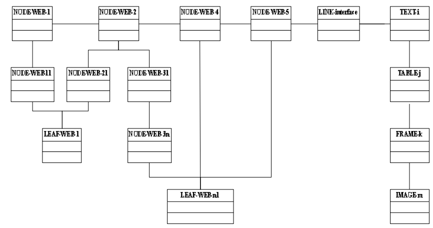

Figure 28: Fourth Class Diagram of Sequential Link-Based Model

Cycle link-based model is one link-based model in which one or more relationships between root web element and node web elements, node web elements, or node web elements and leaf web elements, are loop. There is no ordinal relationship in this link-based model.

The class diagram of cycle link-based model is described in Figure 29.

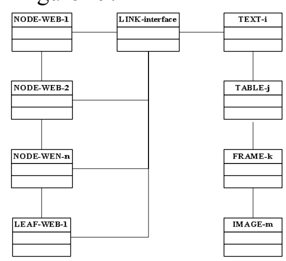

Figure 29: One Class Diagram of Cycle Link-Based Model

Compositive link-based model is one link-based model in which there exist two relationships of loop and ordinal at the same link-based model between root and node web elements, node web elements, or node web elements and leaf program elements.

The class diagram of compositive link-based model is described in Figure 30.

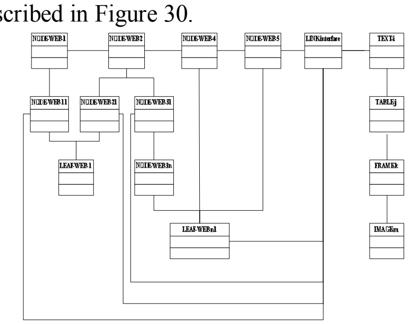

Figure 30: One Class Diagram of Compositive Link-Based Model

The class diagram of cycle link-based model of WEB legacy system is simpler than composite link-based model. If one WEB legacy system has the cycle condition and it means that it has cycle link-based model, then its class diagram is directly acquired from Figure 14 and 29. So as to class diagrams of sequential link-based models of web-based legacy system.

As mentioned above, the use of model for Web-based legacy system improves the modelling efficiency.

Rule 8: Presenting composite structure diagram of five classes.

Composite structure diagram in UML 2.0 depicts the internal structure of a class. Composite structure diagrams, which are new to UML 2.0, focus on instances and their internal structure, providing examples of how the static architecture will achieve a requirement [9, 15]. This diagram is most often used to show hidden internal details of a class, an object, or a component. It is composed of parts and connections.

The text class of web class diagram is composed of font, location, content, period, subscript, superscript, deleted line, formatted text, inserted underline, quoted text, scrolled text, etc (Figure 31).

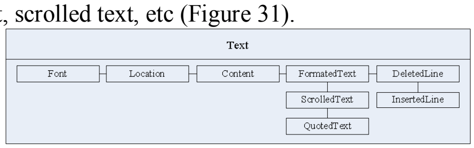

Figure 31: Composite Structure Diagram of Text Class

The image class of WEB contains location, content, embedded image, text substituted with, made map, spacer, etc (Figure 32).

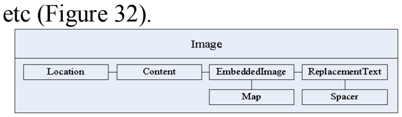

Figure 32: Composite Structure Diagram of Image Class

The table class includes location, table head, table body, table foot, caption, label, selection part, created form part, button, text area, created volume, etc (Figure 33).

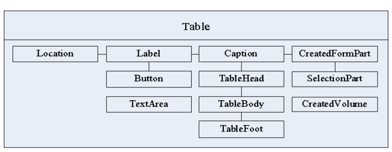

Figure 33: Composite Structure Diagram of Table Class

The frame class includes location, content, embedded frame, frame set, text substituted with, etc (Figure 34).

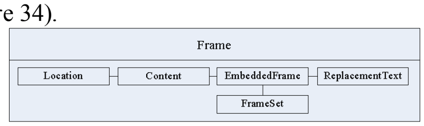

Figure 34: Composite Structure Diagram of Frame Class

The link class includes location, base, layer, linkURLs, etc (Figure 35).

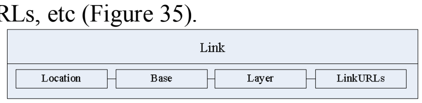

Figure 35: Composite Structure Diagram of Link Class

Sometimes one class is composed of several equivalent parts, and they have the similar structure and the same functions. They are modelled with composite structure diagram in the same way.

Sometimes one part of one class is complicated and has many small pieces. In this case, the internal structure of that part of that class can be modelled with UML composite structure diagram as well.

Rule 9: Presenting component diagram of web legacy system.

A component diagram shows the dependencies among software components, including source code components, binary code components, and executable components [7, 10, 15]. A software module may be represented as a component type. Some components exist at compile time, some exist at link time, and some exist at run time; some exist at more than one time. A component diagram has only a type form, not an instance form. Legacy WEB system depicts the web page. It uses the wet browsers, the database, the application system, etc (Figure 36).

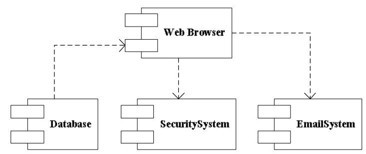

Figure 36: One Component Diagram of Example

Rule 10: Presenting deployment diagram of WEB legacy system.

A UML deployment diagram is the description of the processors, the devices, and the connections between the processors and the devices [1,3]. It shows the hardware for web-based legacy system, the software that is installed on that hardware, and the middleware used to connect the disparate machines to one another.

Web-based legacy system is the presentation of the web usage. It uses the wet browsers, such as Internet Explorer and Netscape, which is allocated in the client, the database, such as FoxPro and Oracle, which is stored in the database server, and the application system, such as the search engine and email, which is deposited in the user client (Figure 37).

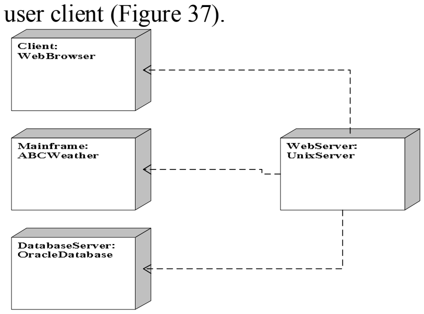

Figure 37: Deployment Diagram of One Example

Rule 11: Applying web rules.

Based on the link-based model of web-based legacy system, four UML diagrams is able to be presented through the above eleven rules. All the discussion above is in the most complex situation. In some cases, one web-based legacy system may not link node web elements, and it only has leaf web elements; one web page may only contain parts of five local web blocks; one web page may only contain several paragraphs of text description; or even it is composed of only one picture. Then those eleven rules do not need to be followed one by one.

So, when one web legacy system has one of the following characters, it is unnecessary to follow those rules:

C1: It is composed of one root element and not more than three leaf web elements.

C2: One web element only has one block.

C3: One web element is composed of several paragraphs of text.

C4: One web element only has one image, such as one picture or one photo.

C5: One of five blocks of one web element has only one class among text class, image class, image class, table class and frame class.

As mentioned in this chapter, Rule 1 to 7 describes the realisation of class diagrams of WEB legacy system. When C1 occurs, which means that this system only has one root web element and not more than three leaf web elements, its class diagram is unnecessary to be realised (Figure 38). When C2 occurs, which means that the web element being discussed only has one block, its class diagram does not need to be realised and Rule 3 to 5 are not used. When C3 happens, which means that the web element only contains text format, its class diagram and composite structure diagram are unnecessary to be realised and Rule 3 to 6 are not used. When C4 happens, which means that the web element is composed of one image, its class diagram and composite structure diagram are unnecessary to be realised. When C5 occurs, which means that the web element only has one block, its class diagram does not need to be realised. Consequently, when conditions C1 to C5 occur, the complexity of modelling web-based legacy system at that point is greatly reduced.

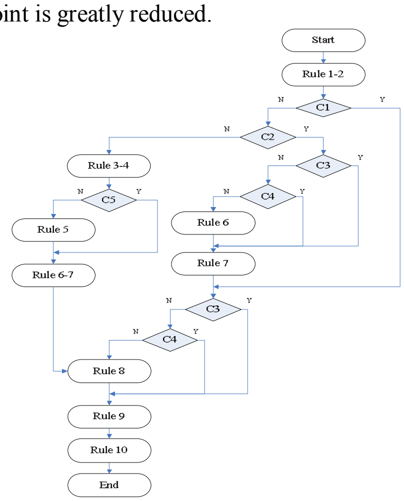

Figure 38: WEB Rules Application

10. Realised Tool and experiments

In order to demonstrate the evolution process, one unified system is realised in this paper, which is called as SEASAT (Software Evolution for domAin-Specific legAcy sysTem). The models of Web-based legacy system correspond to the judgement conditions. SEASAT contains source code file in textual format as the input. It opens one source code file, parses it, searches the models, and displays UML diagram. Its main working flow is presented in Figure 39.

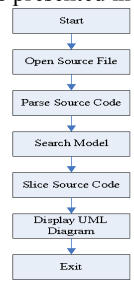

Figure 39: Tool Architecture

SEASAT integrates all technical supports into a systematic method for software evolution of domain-specific legacy systems. It clearly describes the main thoughts of the proposed approach in this paper. Some figures display the main working interfaces in Figure 40 and 41.

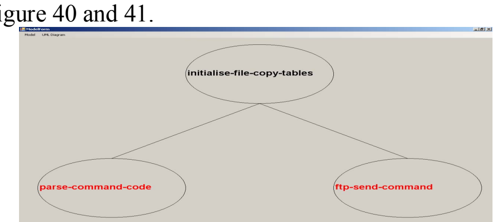

Figure 40: One Link-Based Model of Web Code

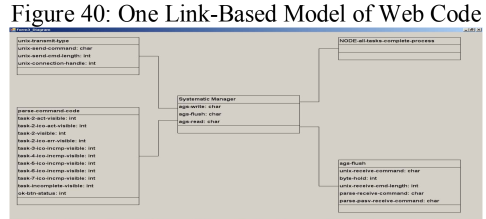

Figure 41: Class Diagram of One Web Code

11. Conclusion

Web-based legacy system has characteristics of presentation of web, data format and basic in usage. After web operations are divided into five groups, the link-based model is introduced. It is one graph to describe the importing or imported relationships of those webs. It has three kinds: sequential, cycle, and compositive link-based model. Eleven rules are presented to realise selected four UML diagrams.

12. References

[1] S. W. Ambler, *The Elements of UML 2.0 Style*, Cambridge University Press, ISBN: 0-521-61667-6, 2005.

[2] J. Conallen, *Building Web Applications with UML*, Addison-Wesley, 2003.

[3] A. Kleppe, J. Warmer and W. Bast, *MDA Explained: The Model Driven Architecture–Practice and Promise*, Addison-Wesley, 2003.

[4] J. Pu, Z. Zhang, R. Millham, Y. Xu and H. Yang, "Modelling Web-Based System with UML Sequence Diagrams", *In the Proceedings of IADIS Virtual Multi Conference on Computer Science and Information Systems (MCCSIS'05),* International Association for Development of the Information Society, 2005.

[5] H. Yang, Advances in UML and XML-Based Software Evolution, Idea Group Pub, 2005.