ConceptBib: A tool for integrating Concept Map and Bibliographic Management

by

Phan Anh Le,

Thesis

Submitted by Phan Anh Le

in partial fulfillment of the Requirements for the Degree of

Bachelor of Software Engineering with Honours (2770)

Supervisor: Dr. David Squire

School of Computer Science and Software Engineering
Monash University

November, 2004

Contents

List of Figures

ConceptBib: A tool for integrating Concept Map and Bibliographic Management

Phan Anh Le,
phanl@csse.monash.edu.au
Monash University, 2004

Supervisor: Dr. David Squire
davids@csse.monash.edu.au

Abstract

There is always demands for more effective of knowledge management (KM) in order to optimize its use. Various efforts have been put forward in research into sound KM. Amongst those is the use of concept maps (CMs) to structure information into the graphical forms showing advantages in representing the relationships amongst illustrated entities. There are also practical applications that can speed up, as well as increase the power and usefulness of KM process. However, there still exists the gap among two fields. We believe that their strengths can be further exploited if they appear in a single tool. This tool will enable users to generate concept maps, to affix references to supporting evidence, and to interactively update the users' reference databases such as BibTEX or EndNote.

ConceptBib: A tool for integrating Concept Map and Bibliographic Management

Declaration

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Phan Anh Le
November 5, 2004

Acknowledgments

I would like to thank my supervisor, Dr. David Squire for the idea of the project, and his patience and guidance throughout the year. Special thanks go to my family and especially my partner, Jenny, for their ongoing support

Phan Anh Le

Monash University
November 2004

Chapter 1

Introduction

There is always demands for more effective of knowledge management in order to optimize its use. Organizing knowledge involves storing information in a suitable medium, structuring information in various ways, and appropriately delivering information to users so that efficient knowledge retrieval, and dissemination are possible. An example of a situation, where knowledge management (KM) is essential, is the literature review process when preparing a scholarly article or thesis. This process requires great amount of reading, note taking, and critical thinking in order to evaluate materials encountered. Moreover, it is vital to identify the key ideas, and the relationships between them, which can be massively complex in large research projects. The challenge is to manage research information properly, from which new knowledge can be derived as well as the correct references made.

1.1 Motivation

There have been various efforts put forward in research into sound KM. Amongst those is the use of concept maps (CM) to structure information into the graphical forms. More specifically, graphs show advantages in representing the relationships amongst illustrated entities; in particular case of KM, the semantic structure of the knowledge can be effectively expressed. CMs can allow us to simplify complex knowledge for which the use of serial textual documents is considered inadequate.

Looking from a more practical point of view, there are applications that can speed up, as well as increase the power and usefulness of KM process. In the particular case of doing research, they are tools that assist researchers in writing papers in general, including typesetters (e.g. LaTeX), and bibliographic management helpers such as BibTeXor EndNote. These tools are greatly helpful in summarizing knowledge that researchers have been across. Despite the advantages these tools provide, the tasks involving doing research review, such as effectively managing ideas and their references, remains a challenge and needs further improvement. Of interest is the possibility of combining the strengths of KM theories and existing

tools to create a more powerful one. If one is provided with better tool, there would be chance that knowledge can be more effectively managed.

1.2 Contribution

By combining CMs and bibliographic management tools, advantages that new tool offers include:

- Modeling tools for organizing knowledge: This research offers more functionalities than a drawing tool does. It aims at providing systematic operations for creating visual knowledge representations, which are more intuitive to remember and more accessible.
- Automated integration between various bibliographic representations: This research also develops a framework, by which the useful conversions and interactions between varied bibliographic formats are possible. This framework was designed so that extensibility is maintained; therefore new potential standards should be supported efficiently.

1.3 Thesis outline

Chapter 1 establishes the introductory information, main goals of the project, as well its motivation. Chapters 2 and 3 reviews strengths and limitations of related knowledge relevant to the project concept. Chapter 4 proposes an approach to problems described above. Chapter 5 details the solutions in terms of technology used, and design structure. Chapter 6 introduces a case study of the use of the tool, ConceptBib, and evaluates potential applications. Chapter 7 concludes the thesis with a brief summary and proposes future work for this project.

Chapter 2

Concept Maps

This section briefly describes the theoretical foundation of CMs, which originated in the field of psychology. This sections also looks at the applications where CMs have performed well, and introduces some practical tools using CMs.

2.1 An overview of concept maps

Jonassen et al. (1993) defined concept maps as "representations of concepts and their interrelationships that are intended to represent the knowledge structure that humans store in their minds". In plain language, concept maps are tools for better management of human knowledge, which help users learn new information more quickly and effectively as they appear in graphical form.

In a less abstract description, a typical concept map is a graph, which consists of two sets: concepts, often represented by named nodes, and the relationships between them or propositions shown by connecting directed arcs. In essence, the concepts are the generalizations of knowledge, of ideas conveyed in any form, e.g. books, documents, speeches, lectures, or observations whereas propositions show how concepts are linked together (Novak; 2002). Intuitively, through a quick look at the map, the authors or the new users would have much more convenience in studying and remembering the semantic contents of the map. The psychological explanation is that the human brain learns and stores information much more efficiently in the graphical forms Ausuel et al. (1978).

The figure 2.1 is an example of a CM, which is a visual description of components commonly found in a simple computer. It is observed that this pictorial representation can be used as a replacement for a textual paragraph detailing the required computer components. Without losing the essential details, the semantics of the computer composition is equivalently captured within a simple yet information-rich concept map. Additionally, one would find the map to be more memorizable than a wordy textual paragraphs.

Figure 2.1: Simple components of a personal computer

To talk briefly about the CMs history, they were introduced as a part of a larger project, dated back in 1960s to 1970s, in which researchers analyzed the development of children's knowledge. In that research, Ausuel et al. (1978) proposed that learning is the integration of new knowledge into learners existing framework, which serves as the foundation for thinking and action. For example, children start to form their cognition from very early ages. As they grow up, their frameworks are gradually extended, deepened, and corrected, since their preliminary perceptions are not always true. Moreover, in order to be stored in long term memory, new knowledge must be semantically connected to others via relationships, rather than being an isolated entity (or rote learning), which can be easily forgotten (Ausuel et al.; 1978).

To model the process of building up a knowledge framework, and to follow the Ausubels research, the definition of concept maps was coined (Novak; 1986). In this theory, building up a knowledge framework is considered to be similar to the creation of a concept map from scratch. The knowledge has inter-relationships, and so do the concepts. By drawing upon good concept maps, educators can benefit in teaching and evaluating what learners have gained from their education (Novak; 1991).

2.2 Applications of Concept Maps:

Although concept maps originated in psychology, they have been used widely in business, engineering, and particularly education. Here we indicate some areas in which the concept map has shown its strengths:

- Facilitating Collaborative Learning: In many areas, efforts invested in learning and creating new knowledge do affect the outcomes. Concept maps can be used in efficiently learning new knowledge. If the shared concept maps are created by groups of learners, they have also been useful in knowledge exchange across individuals as well as larger groups (Novak; 2002).

- Assistance in Knowledge Creation: Because of intuitive nature of concept maps, businesses and engineering corporations have used concept maps in brainstorming new ideas, roadmaps of productions, relationship diagrams, many of which have not been formally recognised as concept maps (J.P. van Schie; 2002).

- Concept Maps for Evaluation: Not only do the concept maps show their effectiveness in knowledge proliferation, they are also a great tool for evaluation of information retained by the learners (Edwards and Fraser; 1983).

2.3 Review of current tools applying Concept Maps

Appying advantages found in CMs, a number of concept map-based applications exist such as FreeMind, ThinkGraph (Thinkgraph Webteam; 2004) , MindMapping (The Mindtools Webteam; 2004), IHMC Concept Map Tools (The IHCM Group; 2004). In regard to the context of this project, we are especially interested in ThinkGraph and IHMC Concept Map Tools because of their popularity and their provided functions.

2.3.1 ThinkGraph software

ThinkGraph is currently a free software for personal and educational use (Thinkgraph Webteam; 2004). It is a systematic tool used mainly for purpose of building up concept maps and summarization of information. Figure 2.2 illustrates a example of a typical concept map created in ThinkGraph.

Advantages of ThinkGraph include easy operations for people to construct concept maps by providing a 2-dimensional work space, where users can quickly create "shapes" of subjects and important links between them. ThinkGraph graphs can be quickly created and extensible they have predefined templates and shapes.

As can be seen from the screenshot in figure 2.2, the map has the tree-based shape, where links are only allowed between a parent and its children. Although this restriction implies the map is hierarchical and more rememberable (because of the levels shown in different colors), it imposes limitations of use on users. There is times when the graph-based approach, which allows links between any components at any level, is needed.

Another limitation is the lack of a mechanism for embedding supporting information in concepts. Technically, ThinkGraph saves the files as Scaleable Vector Graphic (SVG) without adding further information to the format.

2.3.2 IHMC Concept Map Tools

The Institute for Human Machine Cognition (IHMC) Concept Map tool is the product of the research organization (The IHCM Group; 2004) where the definition of concept maps themselves was coined. The IHMC tool is a general purpose tool with no predefined templates, however it empowers users to easily create, navigate, and share concept maps amongst a joint research community.

The strongest advantage of this tool is that it can connect to various registered research sites and retrieve existing concept maps stored in those remote locations.

The figure 2.3 illustrates this function in graphical user interface. This is beneficial if the authors want to search, refer, or integrate the current map into others already shared and standardized by the wider community. In addition, the IHMC tool is platform-independent as it is written in Java. The users can create, modify, and export the concept maps to different formats including image files (such as JPEG, TIFF, and PNG), web pages, and XML file formats at ease.

However, there are still limitations that can be improved from the existing software:

- Limitation in referenced resources: the strong point introduced by IHMC tool is also its limitation. It allows users to connect only to trusted pre-registered hosts, and reference to files stored in those hosts or file store locally. In other words, it limits the number of resources the current concept map can point to.

- Limitations in information embedded in nodes: As it is a general concept maps, users find little support in nesting further information into the nodes. In the current version, it only allows users to enter a few lines of text which will be displayed when users scroll the mouse over specific nodes.

- Limitation in grouping and layering: Current version provides no support for layering or grouping certain types or groups of concepts. These function can be useful in a more complex concept maps where searching and layering are important.

2.3.3 Summary of the tools

Summarizing the review of current tools, there are a number of current limitations in both commercial and educational software that this project aims at partly addressing:

- Hyperlinks and resource locators not fully supported: In the Internet age, it is desirable to be able to quickly access the resources and information supporting a specific concept, not only stored locally or in limited predefined sites. A better solution might include the degree of relevance to the current concept so that the users can select the best suited when they needed. Interaction with other software such as document readers or web browsers would also be useful options.

- Embedded information is limited: Embedded information to concept might be more than simple text. It might include key words used by search engines, references to other resources, multi media files such as images, or audio files. Generally speaking, the embedded information can be pointers to anything that can be used to find supporting or contrasting information of current concepts.

- Classification or layering are not fully supported: As the knowledge grows, graphical representation can be increasingly complex. A function allowing users to quickly identify and group parts of the concept maps would be required.

Figure 2.2: Using ThinkGraph to create tree-shaped concept map.
Source: `http://www.thinkgraph.com`

Figure 2.3: Connect the IHCM tool to pre-registered NASA sites to retrieve an existing Concept Map

Chapter 3

Representation of Bibliographic and Content Metadata

This section is dedicated to looking at management of bibliographic information that can be embedded into the concept maps described in previously. Bibliographic information is generally used to help users to find the exact resources that an author has referred to. This section describes and discusses various methods used to represent bibliographic data.

3.1 BibTEX and its current limitations

BibTeX—A brief introduction:

The creation of TEX by Donald Knuth in the 60s marked the new era of the typesetting of documents, particularly mathematical and technical publications (Kopka and Daly; 1999). TEX offers primitive commands that deal with the simplest formatting functions. Also provided along with TEX is a set of high quality fonts, and even its own processing language (Knuth; 1986).

To continue the success of TEX, the successor, LATEX, which was developed by Leslie Lamport in 1985, is more user-friendly than TEX. While TEX focuses on the formatting and typesetting of documents by issuing low level commands, the main goal of LATEX is to give a set of higher level functions, which are easier for users to use and memorize (Lamport; 1986). LATEX provides markup commands allowing users to easily produce chapters, sections, figures, tables of contents along with others powerful functionalities.

In 1985, Oren Patashnik along side Lamport developed a tool called BibTEX, which processes bibliographic information embedded in LATEX documents. This provides users with an global mechanism of processing citations in LATEX (Kopka and Daly; 1999).

Since then, LaTeX and BibTeX have become amongst the most frequently used tools when researchers write academic papers, particularly in computer science and engineering domains. The mechanism in which LaTeX and BibTeX collaborate is simple yet powerful: LaTeX has the reference keys in its, which point to segments of textual bibliographic data in BibTeX files. After several runs, LaTeX automatically looks at the segments, extracts and integrates them neatly into the referring documents.

Figure 3.1 shows a simple example of a BibTeX entry.

```
@book{Novak1998,
   author= {Novak, J.D.},
   title = {Learning, Creating and Using Knowledge: Concept Maps
             as Facilitative tools in School and Corporation},
   year= {1998},
   publisher={Lawrence Arlbaum and Associates},
 }
```

Figure 3.1: An example of BibTeX entry

3.1.1 Current Limitations:

Although BibTeX has greatly helped writers in processing references since it was created, as requirements of users increased and the types of publication diversified, users started to find a number of limitations in BibTeX. Current limitations of BibTeX which we have observed include:

- Serial presentation and access: Data stored in BibTeXis in plain text therefore it is serially accessed. Serial access can also means the lack of convenience and slowness.

- Lack of standards for online publications: as BiTeX was developed in pre-Internet age, it does not provided formal methods of dealing with online publications.

- Structural issues/cross reference: BibTeX provides 14 listed standard types of citations (Kopka and Daly; 1999). To use any other types, such as images or multi media files, users need to modify style files which are difficult to code, and time consuming.

To solve those limitations, an increasing body of research has proposed tools and data models, most of which are eXtensible Markup Language (XML)-based (The W3C Webteam; 2004). The next coming sections describe these approaches.

3.2 A brief review of EndNote

EndNote is also a bibliographic management suite (The ISI Research Soft; 2000), which is widely used in research community on Microsoft Windows platforms. Noticeable features include:

- Graphical User Interface (GUI): aiming that users who may not be familiar with text-based bibliographic tools, EndNote provides a user-friendly interface with proper layout.
- Connection directly to variety of libraries around the world: EndNote provides collaborative interfaces which enable users can search in pre-registered library collections remotely through EndNote gateways.
- Integrations with Office suites such as one of Microsoft Office: Users can point to reference items easily within the opened editors without switching to EndNote. The results later are inserted following the standard appearance.
- Supports a wide range of bibliographic formats: EndNote offers two native file formats: (i) Older plain text versions, and (ii) More structured XML-based format. Additionally, EndNote can import various popular file formats such as Machine Readable Cataloging (MARC), RISC.
- Strong search capability: Built in EndNote suite are the capabilities of complex search functions that allow users to locate specific resources based on various keys such as the title, authors, subjects.

However, it is observed that EndNote is a commercial product, currently mainly focusing on Windows users and Microsoft platform's software. By doing so, EndNote does not provide support to wider research community, specially computer science and engineering, who enjoy the professional layouts, fonts, and mathematical formulas provided by typesetters such as LaTeX.

3.3 XML Approaches to Bibliographic Data

From the theoretical point of view, the function that BibTeX and EndNote are playing is to contain a higher level of information, that can assist the readers to quickly access the original data. Equivalently, information technology has the term *metadata*, which in plain language means data about other data. Currently, most of the prominent metadata schemes, e.g. Resource Description Framework (RDF), are in XML or XML-related formats since they are portal, easily extensible, well structured, easily understood, and widely supported by applications across domains (Hillman; 2003). By drawing upon technology offered by XML based documents, we can address part of the BibTeX limitations stated previously, including the lack of standards for online based publications, and the structural and cross references issues.

This section looks at a numbers of the most popular representations that have been put forward and used widely in general knowledge management, particularly librarianship.

3.3.1 BibTeXML

To reuse the mass of BibTEX entries and its convenient formatting, a format called BibTeXML has been proposed and used in limited scope. The key idea is that BibTeXML aims at faithfully translating BibTEX entry to a designated XML. An example of a book entry is given in figure 3.2.

```
<bibtex:entry id="kroonenberg">
   <bibtex:book>
      <bibtex:author>Pieter M. Kroonenberg</bibtex:author>
      <bibtex:title>Three-mode principal component analysis</bibtex:title>
      <bibtex:publisher>DSWO Press</bibtex:publisher>
      <bibtex:year>1983</bibtex:year>
      <bibtex:address>Leiden</bibtex:address>
   </bibtex:book>
</bibtex:entry>
```

Figure 3.2: A simple BibTeXML entry

While maintaining faithfully tags of BibTEX help make the format more human readable; however, it limits itself to BibTEX limitations stated earlier. To use it with other open formats, one must define extra elements and rules in provided parsers and translators.

3.3.2 Brief look at Dublin Core

Dublin Core (DC) is the short name for the Dublin Core Metadata Initiative, a project dedicated to promoting the proliferation of interoperable metadata standards and developing more intelligent networked information discovery systems (Powell; 2003). In a much more specific terms, DC aims at developing metadata standards used in across domain, and introducing frameworks that facilitate the use of metadata standards.

Technically, Dublin Core provide a set of XML terms that can be used to effectively describe summary information of the target resource. The highest level element set includes standardized 15 items, and a more comprehensive set of qualified Dublin Core terms (Dublin Core Metadata Initiative; 2003). These elements are

used to semantically cluster information of the subject into subsections for efficient information discovery.

It is worth noticing is that DC element set is not designed for use as a static set, rather than that it provides foundations for much larger and more domain-specific sets of elements defined based on particular requirements. DC introduces an effective standardized guideline, which can be extended or used in conjunction with other body of elements.

3.3.3 Metadata Object Description Schema (MODS)

MODS is introduced by the Library of Congress (LOC) in order to provide a relatively complete set of bibliographic elements that can be used directly in various applications, particularly in librarianship (The Library of Congress; 2004). Being created after Dublin Core and in recognition of limitations of DC applications in specific domains, MODS offers a richer and more complete vocabulary. A complete list of elements and attributes of MODS can be viewed in details at LOC website at *http://www.loc.gov/standards/mods/*

Functionally, MODS can provide as many operations as DC does. Moreover, MODS facilitates the resource descriptions of not only online resources but of any type, ranging from textual contents to multi media such as images, sound recording, or mixed material. Unlike DC as a standard guideline and normally not being used as a standalone model, MODS can be used separately without the further creation of specific body of elements. This is because MODS has strong background support and interrelationships with a range of standards and resource locator systems such as Machine Readable Cataloging (MARC) (The Library of Congress; 2003) or Digital Object Identifier (DOI) (The International DOI Foundation; 2003).

Shown in figure 3.1 is a example of the BibTEX entry and figure 3.3 illustrates the equivalent representation in MODS.It is worth noticing that much more information, such as the subject, the genre of the described resource, resource locators (e.g. URL, DOI, or ISBN), keywords in search engines, access dates, and language information can be easily embedded into MODS, which makes this standard much more open and extensible. (The Library of Congress; 2004)

Looking at these standards from the project perspective, it would be more suitable to apply MODS as it inherits advantages of DC and others, as well as introduces improvements.

```
<mods ID="Novak1998">
    <titleInfo>
        <title>Learning, Creating and using Knowledge</title>
        <subTitle>Concept Maps as Facilitative tools in School and Corporation</subTit
    </titleInfo>
    <name type="personal">
        <namePart type="family">Novak</namePart>
        <namePart type="given">JD</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <originInfo>
        <dateIssued>1998</dateIssued>
        <publisher>Lawrence Arlbaum and Associates</publisher>
    </originInfo>
    <typeOfResource>text</typeOfResource>
    <genre authority="marc">book</genre>
    <identifier type="citekey">Novak1998</identifier>
</mods>
```

Figure 3.3: Bibliographic information in MODS

Chapter 4

ConceptBib: Bridging two fields

Having discussed these two fields above, namely concept maps in psychology, and bibliographical management tools and standards in document publishing, we believe that their strengths can be further exploited if they appear in a single tool. This tool will enable users to generate concept maps, to affix references to supporting evidence, and to interactively update the users' reference databases such as BibTEX or EndNote.

The next sections state the specific requirements for the new tool, discuss an implementation candidate, and propose operations that ensure the extensibility of the project.

4.1 Functional Requirements

The first version of the tool is required to meet the following requirements:

- Usability: Users are not necessarily highly computer literate, and additionally, generally reluctant to learning new complex tools. Therefore, the program must be simple and intuitive to use or, at least, to create simple concept maps
- Embedded Information: It must provide functions that facilitate embedding bibliographic information to the concept maps. Furthermore, a mechanism for interactions between concept maps and bibliographic data must be produced so that the data can be updated accordingly.
- Information retrieval: Apart from representing data, the tool is required to produce techniques so that information about the resources or the resources themselves can be quickly identified and retrieved.
- Extensibility: The first version is currently dealing with BibTEX as the main source of bibliographic data. However, the tool needs to devise a structure to support multiple potential bibliographic formats as well as functional extensions.

4.2 Implementation candidate–Dia

A decision was considered at the beginning of the project where we decided whether to start a new drawing program from scratch or to extend existing ones such as those discussed earlier (The Mindtools Webteam (2004); The IHCM Group (2004)), which support some relevant functionalities. Carefully evaluating their advantages and disadvantages, we decided to select Dia as the application for extension. This section gives a brief description about Dia and its functionalities.

Dia—an open source project available under GNU license—has proven to be a powerful diagram editor used in various application domains, ranging from software development to simple civil engineering designs. Dia provides rich predefined diagram libraries, handling variety of shapes and their related behaviors.

Dia, packaged in major Linux distributions, is popular among the open-source community as it is free, easy to use, multi-purposed, and extensible. The extensibility in Dia, also commonly found in many open source projects, allows users, also as developers, to patch, to extend, and to create additional modules that suit their needs based on an existing stable framework. The framework has been contributed, developed, and tested by many users-developers around the globe.

Considering our project requirements, it needs to provide graphical manipulations with various shapes, points which have already been introduced in Dia. Therefore, an approach was to extend and to utilize Dia to suit our requirements. Our implementation stands as a Dia module or plugin. It inherits common behaviors and properties, but also has its own functions, such as allowing more bibliographic information to be embedded. Potential users need only to drop the compiled binary version to Dia default plugin folders.

Figure 4.1 illustrates how the project communicates with the existing Dia program. On the left hand side is Dia's current core package, which is responsible for diagram drawing, connection point handlers, page layouts, files communications, and others. On the right hand side is our ConceptBib package which offers the requirements mentioned earlier. Two parts communicate via a default interface, a dynamic library of function calls and object registration, offered by Dia for adding new extensions and plugins. ConceptBib, through predefined function calls, can efficiently utilise functions provided by core Dia without much complex knowledge of the internal mechanisms behind the scene. The lower part depicts a important component of the package: A conversion utility which allows ConceptBib to interact with various files formats in outside world.

The reason of choosing Dia as the implementation extension is justified by its strength in graph drawing and its predefined libraries. However, to facilitate further

extension of the this project itself as well as Dia, the package was designed and implemented so that they are as decoupled as possible.

Figure 4.1: High level view of how ConceptBib is integrated with Dia

4.3 eXtensible Markup Language (XML) intermediate level

In effort to allow ConceptBib interactions with various types of bibliographic formats, a model of intermediate XML, based on MODS (The Library of Congress; 2004), was devised. As illustrated in figure 4.1, ConceptBib needs only to directly

interact with the XML; however, it still achieves required portability thanks to a the conversion utility layer. The XML conversion utility translates various file formats to an internal intermediate XML, specifically in MODS, and backwards.

There are two-way conversions considered in this project. Up-stream translating is from original bibliographic files to XML intermediate format. Essentially, in XML information is organized into tree structure which can be attained from various sources, including textual formats or other XML documents, as long as suitable parsers are provided. In up stream translation, most of the bibliographic formats, such as BibTEX or EndNote, are well structured and can be easily parsed. In this project, a BibTEX parser/converter was adopted from Putnam (2004) to provide required conversion to MODS.

The down-stream translating from intermediate XML to original formats can be efficiently accomplished by use of XML Stylesheet Language Transformation (XSLT). In essence, XSLT is a special type of XML, which is instructions used in conjunction with a XML processor to re-organize or completely re-build the XML tree structure. Details about using XSLT in this project is provided in the next section.

Chapter 5

Design and Implementation

During project development, a number of interesting points have emerged. In this section, discussion is dedicated to the project's high-level design, specific Concept objects, and the use of XML and XSLT in relation to requirements mentioned earlier.

5.1 Design Structure

The figure 5.1 illustrates a high level design structure of the project. Keys points worth noticing are as follows:

- Core Dia functionality provides graphic libraries, geometric shapes, connection points and their handlings. These features are offered in Dia diagrams.
- Concept maps are considered to be a specific type of Dia diagrams. A typical concept map contains underlying concepts and propositions, which in turn are represented as rounding boxes or eclipses and directed arrows.
- Concepts inherit drawing capabilities and event handling from Dia objects. Propositions connect concepts together via labeled links.
- Each concept may have one or more reference objects, which contains bibliographic information temporarily in memory parsed from BibTEX or EndNotes files.
- All internal interactions between ConceptBib and bibliographic data are via means of the XML intermediate layer. The final output is converted to native format from XML using XSLT (see the next section on XSTL for more details).
- During the course of editing concept maps, users need to search and send queries to XML files, so the XPath language was employed. XPath provides convenient functions to search for nodes within XML files based on specific queries.

5.2 Concept Object in Dia

The Concept object plays an essential role in the module where the drawing functionalities and embedded information are integrated. As shown in figure 4.1, Concept objects produce calls to Dia's core predefined functions, such as drawing shapes, filling colors, handling events such as mouse clicks, and key strokes. At the same time, they hold the bibliographic information of the represented concepts, calling functions to search for, edit, and save XML entries.

5.3 MODS Entries

As mentioned earlier, XML was chosen to provide a portable medium to support future extension and potential formats. More specifically, in this project, MODS standard was chosen as the XML intermediate level. Although it it is newly proposed and developed by the Library of Congress, it inherits advantages from LOC's existing successful schemes such as Machine Readable Cataloging (MARC) fields commonly used in librarianship. Figure 5.2 show in example of MODS file header. The XML name space attribute specifies the vocabulary, in this case, defined in LOC website.

In this project, we mainly focused on using MODS to sufficiently convey existing and potentially new fields for bibliographic information stored in variety of file formats such as BibTEX, EndNote, Dublin Core, and others. MODS systematically supports various types of resources, including text, images, multimedia, software, and even mixtures of them. Therefore its tags vocabulary is considered sufficiently large, and complex for describing bibliographic information in the foreseeable future.

From implementation point of view, we classified items described in MODS into 2 types, each of which need its own mechanism of extracting and injecting information:

5.3.1 Self-contained Items

These are resources that appear standalone, such as books, theses, manuals. They can have own resource locator, which is used to identify and retrieve information. An example of a book is provided in figure 5.3 (more details of MODS file can be seen in Appendix A).

It is worth noticing few points in figure 5.3:

1. MODS has designated fields for describing persons. Specific roles are defined in MARC Relators such as authors, editors, co-authors, etc...
2. The type of resource is defined in `<genre>` tags.
3. `<identifier type=‘‘citekey’’>` tags can be replaced by or used concurrently with other kind of identifiers such as DOIs, URIs, URLs, ISBNs, etc to extend coverage.

5.3.2 Partial Items

These are resources that are parts of a larger collections of items such as chapters in books, articles in journals, or proceedings material. Consequently, they contain information about the host as follows:

```
<mods ID="Ferguson2003">
    .....
    <relatedItem type="host">
        <titleInfo>
            <title>The Journal of Computing in Small Colleges</title>
        </titleInfo>
        <originInfo>
            <issuance>continuing</issuance>
        </originInfo>
        <genre authority="marc">periodical</genre>
        <genre>academic journal</genre>
        <part>
           <date>2003-Apr</date>
           <detail type="volume"><number>18</number></detail>
           <detail type="number"><number>4</number></detail>
           <extent unit="page">
               <start>344</start>
               <end>354</end>
           </extent>
        </part>
    </relatedItem>
  ....
</mods>
```

It is noticeable that

1. The resource was not directly described in full detail, instead it was injected with embedded information of larger items.

2. The containing item is defined as `<relatedItem>` with type of `host`.

5.4 Conversion using XSLT

XSLT is a language for transforming XML documents into other XML documents or other formats. This could be another XML document, or a document in a different format altogether, such as PDF, HTML, or even plain text (Clark; 1999). XSLT stylesheets contains series of templates which produce the modification or transformation on target elements. One of the most common uses of XSLT is to produce markup tags, which controls how the document appears in viewer or printing, to

a document based on defined rules. For example, one can use XSLT to make the header appears bold and italic. XSLT can also control the order in which elements and attributes are displayed. This means that a new document with different layout, or different structure can be produced.

The use of XSLT is to support portability of outputting the result from ConceptBib. If we have information stored in MODS, it is desireable to convert it to well established formats. Additional XSLT stylesheets could easily be employed to enable the function.

The example shown in Fig 5.4 (more details can be found in Appendix B) illustrates how to extract information about the author from MODS and convert it to BibTEX format:

As can be seen from the listing, XSLT templates consists of series of function-like callings, which test and select appropriate information and output in in desire formatting effect.

5.5 Searching using XPath

Having the XML document in memory, it is necessary to have a means of quickly and systematically searching for desire information stored in complex XML trees. For example, a user might want to retrieve data of a reference entry that has the key of "ABC", or the published year of 2000. Manual traversal through the XML tree is not the optimal solution. A technique for applying XPath in this project was implemented so that users could retrieve information effectively.

Technically, the XML Path Language, a language for addressing parts of an XML document, is designed to be used by both XSLT and XPointer (Clark and DeRose; 1999). The language mainly consists of location paths and expressions. A location path is e.g. `child::para[position=(1)]` that selects the first para child of the current context node. XPath can be used to enquire, select, and filter sub sets of nodes within a XML documents based on the XPath queries. XPath can play the role of a search engine within the scope of the given documents.

An example of XPath queries is as follows:

```
//mods[@ID=‘‘Knuth’’]
```

This says that it should try to find any `mods` nodes that have the ID equal to "Knuth". Multiple criteria can be combined in single XPath queries to to get the desire results:

```
//mods[@ID=‘‘Knuth’ and originalInfo/dateIssued=‘‘1986’’]
```

This query searches for references that have the key equal to "Knuth1986" and published year of 1986.

Figure 5.5 illustrates of how to build an XPath query from user graphical interface (GUI) in ConceptBib. When users clicks on search options, plain text queries such as `id=Knuth&&year=1986` are created. These are fed to XPathQueryBuilder to produce XPath-compliant expressions, which are then executed.

5.6 Universal access and hyperlinks

It is conventional that researchers might keep their own soft copy or pointers to location where information can be quickly retrieved. With aids of URLs, this process is supported transparently to users, either by sending queries to default Internet browsers or call appropriate document viewers, depending on the type of the file.

This project identified 3 ways of online resource retrieval:

1. Local soft copies: This is when users store the soft copies of the files in their local machines, to which they specify pointers in the bibliographic file. The instant viewing can be done by determining the type of the files, possibly by their extension, and invoking default viewers.

2. Uniform Resource Locators (URLs): This is when the files are on remote servers available via various protocols, such as http or ftp. Each available file is assigned a URL used to uniquely identify it over the wider network, and the Internet. Normally, the retrieve calls are sent via Internet browsers, which in turns handle response of the servers. Again, the file type is determined by the web browser, and according to default environment variables, appropriate handlers can be invoked.

3. Digital Object Identifier (DOI), and the like: in the third category are standardized resource locator systems, which allow users to store encoded arrays of characters and/or digits, which are systematically assigned to unique publications, such as books, journal articles, images, multimedia files (Langston and Tyler; 2004). Normally, these identifiers are registered with authorized registration agencies. To retrieve the required resources, there are two steps needed: (i) sending identifiers to managing systems or identifier resolver, *e.g dx.doi.org*, to resolve the current addresses of the resources, (ii) retrieving the resources via the resolved addresses. Amongst advantages of those systems is that end users do not have to update themselves the resource availability. Instead, they only hold the standardized identifiers, leaving the automatic update process to the managing systems. This kind of systems has become increasingly popular recently. Examples include that many publications provided by ACM digital library are assigned DOIs, facilitating more effective resource retrieval.

Figure 5.1: High level view of how classes in ConceptBib

```
<?xml version="1.0" encoding="UTF-8"?>
<modsCollection xmlns=‘‘http://www.loc.gov/mods/v3’’>
<mods ID=‘‘ABC’’>
...
</mods>
```

Figure 5.2: MODS XML file header

```
<mods ID="Lamport1986">
    <titleInfo>
        <title>\\LaTeX</title>
        <subTitle>a document preperation system</subTitle>
    </titleInfo>
    <name type="personal">
        <namePart type="family">Lamport</namePart>
        <namePart type="given">Leslie</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <originInfo>
        <dateIssued>1986</dateIssued>
        <publisher>Addision-Wesley Pub. Co</publisher>
    </originInfo>
    <typeOfResource>text</typeOfResource>
    <genre authority="marc">book</genre>
    <identifier type="citekey">Lamport1986</identifier>
</mods>
```

Figure 5.3: An example of MODS entry

```
<!-- Get a single author -->
    <xsl:template match="name">
        <!--Check the type is personal and played author role -->
        <xsl:if test="@type='personal'">
         <!-- Get the name parts-->
           <xsl:for-each select="namePart">
               <xsl:value-of select="."/>
               <xsl:choose>
                   <xsl:when test="position()=2">
                       <xsl:text>. </xsl:text>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:text>, </xsl:text>
                    </xsl:otherwise>
               </xsl:choose>
          </xsl:for-each>
        </xsl:if>
     </xsl:template>
```

Figure 5.4: Quick example of a XSLT template used to exact author name from MODS

Figure 5.5: Building and executing a XPath query

Chapter 6

Results and Discussion

As the nature of this project is software engineering and the nature of the product itself, evaluating the effectiveness of using ConceptBib would require a comprehensive survey which is described in more depth in the next chapter. Within the scope and the purpose of the project, we evaluated and compared the use of ConceptBib in summarizing our literature review, and also discussed potential use of ConceptBib.

6.1 Case study: Concept Map of this project itself

In figure 6.1, a concept map, built by using the tool, is presented as an example of using ConceptBib. The CM consists of two parts: (i) the left hand side describes the knowledge representation leading to the definition of CMs, and resultant tools, (ii) the right hand side illustrates the bibliographic data representation. The existing CM tools and techniques offer advantages worth reusing, and at the same time, possess a number of limitations, on which ConceptBib can improve. On the other hand, bibliographic and metadata representations, such as BibTeX and DC, are powerful standards offering great deal of applications. However, from the project perspective, MODS, whose element can covers those of BibTeX and DC, shows its strengths, therefore is chosen as the intermediate layer. Traversing further down the map, ConceptBib employes various technology such as XML, XPath, XSLT to maintain its compatibility and extensibility.

Utilizing the memory-friendly characteristics of visual form, one would be able to obtain the semantic flow of the project described in extended textual information more effectively using the map. Furthermore, the references and related information were also embedded so that the resource can be quickly identified as shown in figure 6.2. Additionally, the XPath searching function gives users more functionalities shown in figure 6.3.

6.2 Project outcome

Rather than providing a complete tool, this project developed a framework, on which extensions and potential functions would be possible. Referred back to figure 4.1, noticeable points of the framework are as follows:

1. **Interactions with Dia's core package:** functions for drawing desire shapes, fonts, and formatting options for concepts within the graph were developed. These could be easily adapted to new potential requirements.

2. **Interactions with XML-based MODS entries** functions for building XPath queries systematically, and executing queries were also introduced.

3. **Conversion to other bibliographic formats:** conversions could be extended by adding new XSLT stylesheet files.

6.3 Potential Applications

Potential applications might include:

- **Use of tools for research scientist in managing their large reference collections:** If the tool was to be used routinely, researchers would have a visual representation, summarizing knowledge they have been across. Moreover, it would provide more effective resource retrieval thanks to the hyper link functionalities.

- **A facilitative tool for better knowledge sharing among workgroups:** Research groups would maintain the database of concept maps following some guidelines. This would be helpful in reviewing current knowledge, leading to new ideas created. When new members join the group, concept maps might play the role of the introduction, helping them familiarize faster in the group.

Figure 6.1: Concept Map of this project

Figure 6.2: Embedding information to the Concept Map of this project

Figure 6.3: Searching information using XPath in the Concept Map of this project

Chapter 7

Conclusion

During this project, a comprehensive review of CMs and bibliographic data was created. This review specified that the fields mentioned earlier have their own advantages. However, current tools show limited facilities to optimize and specialize their use in doing research. The requirement that CMs and open bibliographic data should be combined in single tools was driving force for the model design.

Once the requirements were drawn, a model and design structure was constructed. For the purpose of code reuse and future extension, a Dia plugin was chosen as development approach. The design was also carefully created so that the extensibility must be maintained.

After the design was created, an initial version of software package was implemented. Although not supporting complete set of bibliographic files, this program demonstrated the model that we trying to achieve is feasible.

Using the new program, a case study, which was graphical flow of the project information itself, was introduced. It demonstrated that the knowledge can be efficiently conveyed, and be more intuitive then serial textual format. Some thoughts of future usability surveys was also put forward.

7.1 A future survey

Because of resources required in a comprehensive survey on effectiveness of the tool for students when doing there research, particularly in their literature review, we had to leave it as future work or an extension of the project. However, some preliminary thoughts were put forwards in outlining the survey:

- New students, whose knowledge on the specific areas would be even, joining a same research group are divided into 2 groups. Those in group 1 will be given a ConceptBib, with predefined concept maps, while others in group 2 will not.

Both groups will be given access to the same reference database, maintained by the group and recommended by members.

- Tests, used to evaluate the understandings of students, are to be developed and given to them periodically. Scoring scheme also would need to be introduced to quantify the effectiveness.
- Based on the scores, feedbacks and performance obtained by students' answers, effectiveness of using ConceptBib can be then analyzed and summarized.

7.2 Future extension

This tool would be extended or further researched so that more functionalities would be offered. Some of them are:

- **Layering view:** this function might be offered to view or to filter subset of the graphs based on the certain rules of relationships among concepts. This will help users to focus better on specific parts of the map.
- **Include more supported file formats:** The initial version of ConceptBib supports only BibTEX. However, potential formats (e.g. EndNote) can be integrated by following the description provided earlier.
- **Serialization the CMs:** Algorithms to select starting and ending concepts, then to view the logical path linking the tool can be added. This would help users in determining a path to go through in writing literature review.
- **Better searching and hyper link options:** Sending searching keywords and queries to database or search engines to retrieve referred resources might be included. This might involve filtering the results based on their relevance to the target concept.

Appendix A

An Example of MODS files

```
<?xml version="1.0" encoding="UTF-8"?>
<modsCollection xmlns="http://www.loc.gov/mods/v3">
<mods ID="Getty90">
    <titleInfo>
        <title> The X Window System, version 11</title>
    </titleInfo>
    <name type="personal">
        <namePart type="family">ever</namePart>
        <namePart type="given">Who</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <name type="personal">
        <namePart type="family">Collins</namePart>
        <namePart type="given">Phi</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <name type="personal">
        <namePart type="family">James</namePart>
        <namePart type="given">Scott</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <typeOfResource>text</typeOfResource>
    <relatedItem type="host">
        <titleInfo>
            <title>Software Practice and Experience</title>
        </titleInfo>
        <originInfo>
            <issuance>continuing</issuance>
        </originInfo>
        <genre authority="marc">periodical</genre>
        <genre>academic journal</genre>
        <part>
           <date>1990</date>
           <detail type="volume"><number>20</number></detail>
           <detail type="number"><number>S2</number></detail>
        </part>
    </relatedItem>
    <abstract> A Techical Overview of X11 Functionality </abstract>
```

```
    <identifier type="citekey">Getty90</identifier>
</mods>
<mods ID="Novak1998">
    <titleInfo>
        <title>Learning, Creating and using Knowledge</title>
        <subTitle>Concept Maps as Facilitative tools in School and Corporation</
    </titleInfo>
    <name type="personal">
        <namePart type="family">JD</namePart>
        <namePart type="given">Novak</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <originInfo>
        <dateIssued>1998</dateIssued>
    </originInfo>
    <typeOfResource>text</typeOfResource>
    <identifier type="citekey">Novak1998</identifier>
</mods>
<mods ID="Hillmann2003">
    <titleInfo>
        <title>Using Dublic Core</title>
    </titleInfo>
    <name type="personal">
        <namePart type="family">D</namePart>
        <namePart type="given">Hillman</namePart>
        <role>
            <roleTerm authority="marcrelator" type="text">author</roleTerm>
        </role>
    </name>
    <typeOfResource>text</typeOfResource>
    <relatedItem type="host">
        <titleInfo>
            <title/>
        </titleInfo>
        <originInfo>
            <issuance>continuing</issuance>
        </originInfo>
        <genre authority="marc">periodical</genre>
        <genre>academic journal</genre>
        <part>
           <date>2003</date>
        </part>
```

```
    </relatedItem>
    <identifier type="citekey">Hillmann2003</identifier>
</mods>
</modsCollection>
```

Appendix B

Part of the XSLT stylesheet

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="text"/>
    <xsl:template match="/">
        <xsl:apply-templates select="/modsCollection/mods"/>
    </xsl:template>
    <xsl:template match="/modsCollection/mods">
        <xsl:apply-templates select="./genre"/>
        <xsl:value-of select="@ID"/>
        <xsl:apply-templates select="./titleInfo"/>
        <!-- Get author(s) -->
        <xsl:text>,&#10;    author={</xsl:text>
        <xsl:for-each select="./name">
            <xsl:apply-templates select="."/>
            <xsl:choose>
                <xsl:when test="position()!=last()">
                <xsl:text> and </xsl:text>
                </xsl:when>
                <xsl:otherwise><xsl:text>}</xsl:text></xsl:otherwise>
            </xsl:choose>

        </xsl:for-each>
        <xsl:apply-templates select="./originInfo/publisher"/>
        <xsl:apply-templates select="./relatedItem"/>
        <xsl:text>&#10;}&#10;</xsl:text>

    </xsl:template>

    <!-- Get the type -->
    <xsl:template match="/modsCollection/mods/genre">
        <xsl:text>@</xsl:text>
        <xsl:value-of select="."/>
        <xsl:text>{</xsl:text>
    </xsl:template>
    <!--Get Title and subTitle -->
    <xsl:template match="/modsCollection/mods/titleInfo">
        <xsl:text>,&#10;    title={</xsl:text>
        <xsl:value-of select="./title"/>
        <xsl:text> </xsl:text>
        <xsl:apply-templates select="./subTitle"/>
        <xsl:text>}</xsl:text>
    </xsl:template>
```

```
<xsl:template match="/modsCollection/mods/titleInfo/subTitle">
    <xsl:value-of select="."/>
 </xsl:template>

<!-- Get a single author -->
<xsl:template match="name">
    <!--Check the type is personal and playing author role -->
    <xsl:if test="@type='personal'">

       <xsl:for-each select="namePart">
           <xsl:value-of select="."/>
           <xsl:choose>
               <xsl:when test="position()=2">
                   <xsl:text>. </xsl:text>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:text>, </xsl:text>
                </xsl:otherwise>
           </xsl:choose>
      </xsl:for-each>
    </xsl:if>
 </xsl:template>

<xsl:template match="publisher">
    <xsl:text>,&#10;    publisher={</xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>}</xsl:text>
</xsl:template>

<xsl:template match="relatedItem">
<xsl:text>,&#10;    journal={</xsl:text>
<xsl:value-of select="./titleInfo/title"/>
<xsl:text>}</xsl:text>
<xsl:apply-templates select="./part/detail"/>
</xsl:template>

<xsl:template match="detail">
    <xsl:text>,&#10;    </xsl:text>
    <xsl:value-of select="@type"/>
    <xsl:text>={</xsl:text>
    <xsl:value-of select="./number"/>
    <xsl:text>}</xsl:text>
</xsl:template>
```

```
</xsl:stylesheet>
```

References

Ausuel, D., Novak, J. and Hanesian, H. (1978). *Educational Psychology, A Cognitive View*, Hold, Reschart and Winstons.

Clark, J. (1999). XSL transformation. Access 26–07–2004.
URL: *http://www.w3.org/TR/xslt*

Clark, J. and DeRose, S. (1999). XPath language. Access 26–07–2004.
URL: *http://www.w3.org/TR/xpath*

Dublin Core Metadata Initiative (2003). DCMI Metadata Terms. : Accessed 10–07–2004.
URL: *http://www.dublincore.org/documents/dcmi-terms/*

Edwards, J. and Fraser, K. (1983). Concept maps as reflectors of conceptual understanding, *Research in Science Education* **13**: 19–26.

Hillman, D. (2003). Using Dublin Core, *Using Dublin Core Web Site* . Access date: 26–03–04.
URL: *http://dublincore.org/documents/usageguide/*

Jonassen, D., Beissener, D. and Yacci, M. (1993). *Structural knowledge: Techniques for representing, conveying and acquiring structural knowledge*, Erlbaum, Hillsdale(N.J).

J.P. van Schie (2002). Concept Mapping Visualization Tools for Knowledge Management. Accessed 10–07–2004.
URL: *http://www.observetory.com/conceptmappingvs1.htm*

Knuth, D. (1986). *The TeXbook*, Addision-Wesley.

Kopka, H. and Daly, P. (1999). *A Guide to LaTeX: Document Preparation for Beginners and Advanced Users*, 3rd edn, Pearsion Education Ltd.

Lamport, L. (1986). *LaTeX: a document preperation system*, Addision-Wesley Pub. Co.

Langston, M. and Tyler, J. (2004). Linking to journal articles in an online teaching environment: The persistent link, doi, and openurl, *The Internet and Higher Education* **7**: 51–58.

Novak, J. (1986). *Introduction to Concept Mapping: A Handbook for Educators*, Ithaca, Cornell University, New York.

Novak, J. (1991). Clarify with concept maps, *The Science Teacher* **58(7)**: 45–49.

Novak, J. (2002). The theory underlying concept maps and how to construct them, *Website of Institute for Human & Machine Cognition (IHMC)* . Accessed Date:13–04–2004.
URL: *http://cmap.coginst.uwf.edu/info/*

Powell, A. (2003). Expressing Dublin Core in HTML/SHTML meta and link element, *DCMI Official Website* . Access Date: 26–03–2004.
URL: *http://dublincore.org/documents/dcq-html/*

Putnam, C. (2004). Bibliography conversion utilities.
URL: *http://www.scripps.edu/ cdputnam/software/bibutils/bibutils.html*

The IHCM Group (2004). IHCM Concept Map Tool. Accessed 10–07–2004.
URL: *http://cmap.ihmc.us*

The International DOI Foundation (2003). DOI Overview.
URL: *http://www.doi.org/*

The ISI Research Soft (2000). *EndNote: the All-in-One solution*, Berkeley, Calif: ISI ResearchSoft.

The Library of Congress (2003). MARC Standards.
URL: *http://www.loc.gov/marc*

The Library of Congress (2004). Outline of Elements and Attributes in MODS Version 3.0. : Accessed 10–07–2004.
URL: *http://www.loc.gov/standards/mods/v3/mods-3-0-outline.htm*

The Mindtools Webteam (2004). MindMaps - A Powerful Approach to Note Taking, *MindTools Website* . Access Date: 10–07–2004.
URL: *http://www.mindtools.com/pages/article/newIS_01.htm*

The W3C Webteam (2004). Extensible Markup Language (XML). Access 26–07–2004.
URL: *http://www.w3.org/XML/*

Thinkgraph Webteam (2004). Introduction to thinkgraph.
URL: *http://www.thinkgraph.com/english/index.htm*