CHAPTER 2

LITERATURE REVIEW

This chapter presents theoretical background related to this research. Literature relating to document technology, background of metadata, approaches for metadata generation, Information Extraction (IE), Case-based Reasoning (CBR) and Knowledge Management Systems (KMS) will be presented.

The chapter is organized as follows. Section 1 discusses issues related to document technology. Section 2 discusses issues related to background of metadata. Section 3 discusses issues related to approaches for metadata generation. The review of document technology, metadata and approach for metadata generation are to investigate the feasibility of using metadata to provide a mean to standardize conceptualization of knowledge and to facilitate well-defined domain knowledge in community of practice. Section 4 discusses issues related to concept and architecture of IE. The review of IE is to investigate the feasibility of using Natural Language Processing (NLP) and Information Extraction (IE) techniques to allow knowledge sharing between librarians and domain experts. Section 5 discusses CBR and the four phases of CBR cycles. The review of CBR is to investigate the feasibility of using techniques in CBR to allow knowledge sharing and reuse to be achieved. Section 6 discusses issues related to KMS and development of KMS. The review of KMS is to

investigate the feasibility of developing the web-based KMS to allow application software to be implemented. Conclusion is followed in Section 7.

2.1 Document Technology

This section presents character set standard, markup language and stylesheets format and web-based document technology.

2.1.1 Character Set Standard

The most important consideration concerning character sets is standardization. Transferring text between different makes of computer, interfacing peripheral devices from different manufacturers and communicating over networks are everyday activities. Continual translation between different manufacturers' character codes would not be acceptable, so a standards is necessarily somewhat dry, but and understanding of them is necessary if you are to avoid the pitfalls of incompatibility and the resulting corruption of texts. Unfortunately, standardization is never a straightforward business, and the situation with respect to character codes remains somewhat unsatisfactory.

In Thai language, the only standard for Thai character set for information interchange is TIS 620-2533 (1990) which is defined by the Thai Industrial Standard Institute (TISI), Ministry of Industry. It defines two eight-bit character code sets by

extending ISO 646-1983 and IBM GX20-1850-4 (EBCDIC), respectively. TIS 620-2533 (1990) is indeed a minor correction of the previous one, TIS 620-2529 (1986).

2.1.1.1 ASCII

American Standard Code for Information Interchange (ASCII) is the dominant character set from the 1970s into the early twenty-first century. It uses 7 bits to store each code value, so there is a total of 128 code points. The character repertoire of ASCII only comprises 95 characters, however. The values 0 to 31 and 127 are assigned to control characters, such as form-feed, carriage return and delete, which have traditionally been used to control the operation of output devices. The control characters are a legacy from ASCII's origins in early teletype character sets. Many of them no longer have any useful meaning, and are often appropriated by application programs for their own purpose.

American English is one of the few languages in the world for which ASCII provides an adequate character repertoire. Attempts by the standardization bodies to provide better support for a wider range of languages began when ASCII was adopted as an ISO standard (ISO 646) in 1972. ISO 646 incorporates several national variants on the version of ASCII used in the United States, to accommodate, for example, some accented letters and national currency symbols.

A standard with variants is no real solution to the problem of accommodating different languages. If a file prepared in one country is sent to another and read on a

computer set up to use a different national variant of ISO 646, some of the characters will be displayed incorrectly. For example, a hash character (#) typed in the United States would be displayed as pound sign (L) in the UK (and vice versa) if the British user's computer used the UK variant of ISO 646. (More likely, the hash would display correctly, but the Briton would be unable to type a pound sign, because it is more convenient to use US ASCII (ISO 646-US) anyway, to prevent such problems.)

A better solution than national variants of the 7-bit ISO 646 character set lies in the provision of a character set with more code points, such that the ASCII character repertoire is mapped to the value 0-127, thus assuring compatibility, and additional symbols required outside the USA or for specialized purposes are mapped to other values. Doubling the set of code points was easy: the seven bits of an ASCII character are invariably stored in an 8-biy byte. It was originally envisaged that the remaining bit would be used as a parity bit for error detection. As data transmission became more reliable, and superior error checking was built in to higher-level protocols, this parity bit fell into disuse, effectively becoming available as the high-order bit of and 8-bit character.

Predictably, the different manufacturers each developed their own incompatible 8-bit extensions to ASCII. These all shared some general features: the lower half (code points 0-127) is identical to ASCII; the upper half (code points 128-255) held accented letters and extra punctuation and mathematical symbols. Since a set of 256 values is insufficient to accommodate all the characters required for every alphabet in

use, each 8-bit character code had different variants; for example, one for Western European languages, another for languages written using the Cyrillic script, and so on.

Despite these commonalities, the character repertoires and the code values assigned by the different manufacturers' character sets are different. For example, the character e' (e with an acute accent) has the code value 142 in the Macintosh Standard Roman character set whereas it has the code value 233 in the corresponding Windows character set, in which 142 is not assigned as the value for any character; 233 in Macintosh Standard Roman, on the other hand, is E'. Because the repertoires of the character sets are different, it is not even always possible to perform a translation between them, so transfer of text between platforms is problematical. Clearly, standardization of 8-bit character sets was required. During the 1980s the multi-part standard ISO 8859 is produced. This defines a collection of 8-bit character set, each designed to accommodate the needs of a group of languages (usually geographically related). The first part of the standard, ISO 8859-1, is usually referred to as ISO Latin1, and covers most Western European languages. Like all the ISO 8859 character sets, the lower half of ISO Latin1 is identical to ASCII (i.e. ISO 646-US); the code points 128-159 are mostly unused, although a few are used for various diacritical marks.

In Thai language, There are 87 letters in total, including consonants, vowel signs, tone marks, symbols, and Thai digits. Basically, the upper part of the 8-bits ASCII table is large enough to provide a code point to each letter. Due to the past

font and rendering technology, extensional code points are assigned to some glyphs to overcome the rendering and printing problems.

2.1.1.2 Unicode ISO 10646

The only possible solution to the problem of insufficient code points is to use more than one byte for each code value. A 16-bit character set has 65,536 code points, it can accommodate 256 variants of an 8-bit character set simultaneously. Similarly, a 24-bit character set can accommodate 256 16-bit character sets, and a 32-bit character set can accommodate 256 of those. ISO (in conjunction with the IEC) set out to develop a 32-bit Universal Character Set (UCS), designated ISO 10646, structured in this way: a collection of 232 characters can be arranged as a hypercube (a four-dimension cube) consisting of 256 groups, each of which consists of 256 planes of 256 rows, each comprising 256 characters (Which might be the character repertoire of an 8-bit character set). The intention was to organize the immense character repertoire allowed by a 32-bit character set with alphabets distributed among the planes in a linguistically sensible way, so that the resulting character set would have a clear logical structure.

Unicode provides code values for all the characters used to write contemporary "major" languages, as well as the classical forms of some languages. The alphabets available include Latin, Greek, Cyrillic, Armenian, Hebrew, Arabic, Devanagari, Bengali, Gurmukhi, Gujarati, Oriya, Tamil, Telugu, Kannada, Malayalam, Thai, Lao, Georgian and Tibetan, as well as the Chinese, Japanese and Korean ideograms and the

Japanese and Korean phonetic and syllabic scripts. Unicode also includes punctuation marks, technical and mathematical symbols, arrows and the miscellaneous symbols usually referred to as “dingbats” (pointing hands, stars, and so on). In addition to the accented letters included in many of the alphabets, separate diacritical marks (such as accents and tildes) are available and a mechanism is provided for building composite characters by combining these marks with other symbols. (This not only provides an alternative way of making accented letters, it also allows for the habit mathematicians have of making up new symbols by decorating old ones.)

In Unicode, code values for nearly 39,000 symbols are provided, leaving some code points unused. Others are reserved for the UTF-16 expansion method (described briefly later on), while a set of 6400 code points is reserved for private use, allowing organizations and individuals to define codes for their own use. Even though these codes are not part of the Unicode standard, it is guaranteed that they will never be assigned to any character by the standard, so their use will never conflict with any standard character, although it might conflict with those of other individuals.

Unicode is restricted to characters used in text. It specifically does not attempt to provide symbols for music notation or other symbolic writing systems that do not represent language.

Unicode and ISO 10646 were brought into line in 1991 when the ISO agreed that the plane (0, 0, *, *), known as the Basic Multilingual Plane (BMP), should be identical to Unicode. ISO 10646 thus utilizes CjK consolidation, even though its 32-

bit code space does not require it to do so. The overwhelming advantage of this arrangement is that the two standards are compatible (and the respective committees have pledged that they will remain so). To understand how it is possible to take advantage of this compatibility, we must introduce the concept of a character set encoding.

An encoding is another layer of mapping, which transforms a code value into a sequence of bytes for storage and transmission. When each code value occupies exactly one byte it might seem that the only sensible encoding is an identity mapping where each code value is stored or sent as itself in a single byte. Even in this case, though, a more complex encoding may be required. Because 7-bit ASCII was the dominant character code for such a long time, there are network protocols which assume that all character data is ASCII and remove or mangle the top bit of any 8-bit byte. To avoid this it may be necessary to encode 8-bit characters as sequences of 7-bit characters.

One encoding used for this purpose it called Quoted-Printable (QP). This works quite simply – any character with a code in the rage 128-255 is encoded as a sequence of three bytes. The first is always the ASCII code for =; the remaining two are the codes for the hexadecimal digits of the code value. For example, e' has value 233 in ISO Latin1, which is E9 in hexadecimal, so it is encoded in QP as the ASCII string = E9. Most characters with codes less than 128 are left alone. An important exception is = itself, which has to be encoded, otherwise it would appear to be the first byte of the encoded version of some other character. Hence, = appears as = 3D.

For ISO 10646 the obvious encoding scheme, known as UCS-4, employs four bytes to hold each code value. Any value on the BMP will have the top two bytes set to zero. Since most values that are currently defined are on the BMP, and since economic reality suggests that for the foreseeable future most characters used in computer systems and transmitted over networks will be on the BMP, the UCS-4 encoding wastes space. ISO 10646 therefore supports an alternative encoding, UCS-2, which drops the top two bytes. UCS-2 is identical to Unicode.

Unicode encoding goes further. There are three UCS Transformation Formats (UTFs) which can be applied to Unicode code values. UTF-8 takes the reasoning we just applied to 32-bit values a step further. ASCII code values are likely to be more common in most text than any other values. Accordingly, UTF-8 encodes UCS-2 values so that if their high-order byte is zero and the low-order byte is less than 128, the value is encoded as the single low-order byte. That is, ASCII characters are represented by the same value in ASCII and UTF-8. Otherwise, the two bytes of the UCS-2 value are encoded using up to six bytes, with the highest bit of each byte set to 1 to indicate it is part of an encoded string and not an ASCII character. (This means that the UTF-8 encoding of characters which are not in the ASCII repertoire is not the same as their encoding in ISO Latin1.)

Text encoding with UTF-8 is a string of 8-bit bytes, and is therefore vulnerable to mangling by protocols that can only handle ASCII. UTF-7 is an alternative encoding which uses a technique similar to that described for QP to turn Unicode characters into streams of pure ASCII text, which can be transmitted safely.

The UTF-16 encoding has a different emphasis. This encoding allows pairs of 16-bit values to be combined into a single 32-bit value, thus extending the repertoire of Unicode beyond the BMP. Only values in a limited range can be combined this way, with the result that UTF-16 only provides access to an additional 15 planes of the full ISO 10646 character set. These comprise nearly a million characters under UTF-16, which seems to be sufficient for present purpose.

The markup languages used on the web use UTF-8 Unicode as their character set by default, as do Java and other popular programming languages. Unicode has also been adopted as the native character set in recent versions of the major operating systems, so it is now much easier than it used to be transfer text that uses characters beyond the US-Anglophone ASCII repertoire.

Therefore, Thai Unicode provides a unique number for every character, no matter what the platform, no matter what the program, no matter what the language. The Thai Unicode Standard was adopted by such industry leaders as Apple, HP, IBM, JustSystems, Microsoft, Oracle, SAP, Sun, Sybase, Unisys and many others. Thai Unicode is required by modern standards such as XML, Java, LDAP, CORBA 3.0, WML, etc., and is the official way to implement ISO/IEC 10646. It is supported in many operating systems, all modern browsers, and many other products. The emergences of the Unicode Standard, and the availability of tools supporting it, are among the most significant recent global software technology trends.

2.1.1.3 Classification of Fonts

There are thousands of fonts available, each with its own special personality. Some broad characteristics are used to help classify them. A major distinction is that between monospaced (or fixed-width) and proportional fonts.

In a monospaced font, every character occupies the same amount of space horizontally, regardless of its shape. This means that some letters have more white space around them than others. For example, the narrow shape of a lower-case l must be surrounded with white to make it the same width as a lower-case m. In contrast, in a proportional font, the space each letter occupies depends on the width of the letter shape. Paradoxically, this produces a more even appearance, which is generally felt to be easier to read in most contexts. It also allows you to fit more words on to a line. Text in a monospaced font looks as if it was produced on a typewriter or a teletype machine. It can sometimes be used effectively for headings, but monospaced text is especially suitable for typesetting computer program listings. It is also useful for conveying a "low-tech" appearance, for example in contexts where you wish to convey a slightly informal impression. Probably the most widely used monospaced font is Courier, which was originally designed for IBM typewriters in the 1950s, and has achieved wide currency because it is one of the fonts shipped as standard with all PostScript printers. Text in a proportional font has the appearance of the printed text in a traditional book, and is usually preferred for setting lengthy texts. It is generally felt to be more readable, since letters appear to be tightly bound together into words.

In Thai font, the multiple levels of Thai texts are rendered by means of combining characters. WTT 2.0 classifies printable characters in TIS 620 as forward characters and dead characters. Forward characters are those characters written on the base line and occupy horizontal space, while dead characters are those written below or above base line and whose widths are zero. Thai quality display and printing has been enabled by the font technologies. Among these, TrueType is the most widely used. New codes for presentation purpose have been added. For example, since some Thai consonants with long tails occupy a little space in the upper level, new codes for upper vowels and tone marks shifted left have been added. Another need is the adjustment of tone mark position when upper vowel is absent. Thus, new codes for lower tone marks are also added. However, there are two major different code tables for Thai TrueType fonts. One is defined for Mac OS Thai, another is for Microsoft Windows. The one for Mac OS is based on MacThai character set, while that for Microsoft is an extension of code page. This makes the two kinds of TrueType fonts unable to be used together.

2.1.2 Markup and Stylesheets

In the days before desktop publishing and word processors, authors' manuscripts were usually produced on a typewriter. Consequently, they could not be laid out in exactly the same form as the final published book or paper. The author, the publisher's copy editor and possibly a book designer would annotate the manuscript, using a variety of different colored pens and a vocabulary of special symbols, to indicate how the text should be formatted by the typesetter when it was eventually

printed. This process of annotation was called “marking up” and the instructions themselves were often referred to as markup. With the transition to computer-based methods of publication, the function of markup was transferred to annotations inserted (often by the author) into the digital text file that now corresponds to the manuscript. At first, markup was used to specify the detailed appearance of the text, as it had been before computer programs were used, but markup is now generally used to indicate the structure of a document, with its appearance being specified separately.

Many text documents are prepared using WYSIWYG formatting systems and word processors. WYSIWYG stands for “What you see is what you get”, a phrase which succinctly capture the essence of such systems. As you type, text appears on the screen laid out just as it will be when it is printed or displayed. Font and size change, indentation, tabulation and other layout features are controlled by menus, command keys or toolbar icons, and their effect is seen immediately. The markup, although present, is invisible. Only its effect can be seen, presenting the illusion that the formatting commands do not insert markup, but actually perform the formatting before your very eyes, as it were.

An alternative way of working it to prepare the document using a plain text editor. In this case the text is interspersed with the markup, which takes the form of special layout commands, often known as tags. Tags are lexically distinguished from the text proper by being enclosed in angle brackets of beginning with a backslash character, for example. (Text editors often have modes doe common markup languages, in which the tags can be inserted using a single keystroke or menu

selection, so inserting tag-based markup is no more arduous than applying WYSIWYG formatting.) Tags do the same job as the commands in a WYSIWYG system, but their effect is not immediately visible. A separate processing phase is usually required, during which the formatting described by the tags is applied to the text, which is displayed (in a Web browser, for instance) or converted into a form that can be displayed or printed, such as PDF.

Although the subjective experience of preparing text using these two different types of system is very different, calling for different skills and appealing to different personalities, their differences are mostly superficial and concern the interface much more than the way in which layout is actually controlled. Underneath any WYSIWYG system is a tag-based text formatter; sometimes the tags may be binary control codes which are inaccessible to the user of the WYSIWYG editor, but they may be text tags which can be accessed using a different mode or by opening the file in a text editor.

For the most part, we will concentrate on layout using textual tags, partly because it is easier to see what is going on, and partly because, after years in the shade of layout based on hidden control codes, it has acquired a new importance because of the World Wide Web. An advantage of using textual markup, instead of binary codes or some data structure is that the marked up document is plain text, which can be read on any computer system and can be transmitted unscathed over a network. This is particularly important for the Internet, with its heterogeneous architectures. Here, HTML brought tagging into wide use. XML, and a clutch of languages developed

from it, promised to extend the use of this sort of markup into other areas of multimedia, but so far there has been no widespread acceptance of XML as a broad basis for markup. In Chapter 15 we will look at XML and describe how its tagging mechanism has been applied to vector graphics in SVG.

The difference between tag-based text editing and WYSIWYG layout is merely a cosmetic difference between interfaces. A more profound distinction is that between visual markup and structural markup, which we touched on when describing character styles. In visual markup, tags or commands are used to specify aspects of the appearance of the text, such as fonts and type sizes. In structural markup, tags identify logical elements of a document, such as headings, lists or tables, and the visual appearance of each type of element is specified separately.

The distinction between visual and structural markup exists in crude form in any word processor that supports the use of paragraph and character styles. For instance, if you were writing a paper divided into sections, you would almost certainly want to distinguish the section headings from the body text. A purely visual approach to markup would lead you to select each heading, insert space above it and set the font, its size and shape, using the appropriate menu commands or controls on the formatting palette. You would have to apply the correct setting to each section heading.

Commonly, many documents will share a similar appearance. For instance, the headings on each page in a Web site should all be styled in the same way, to give an

appearance of unity to the site. This can be done by putting a set of rules in a separate document, called a stylesheet, which is attached to every page in the site. For each tag being used, a stylesheet provides one or more rules describing the way in which elements with that tag should be laid out. Rules may specify styling in a context-dependent way, so that, for instance, a list element may be formatted differently when it occurs in the header of a page and when it occurs in the main body. There may be more than one stylesheet for a particular document or collection of documents, providing a different appearance to the same structure as a fancy style with extravagant graphic design and a plain style intended for people with poor eyesight. Stylesheets can also be used to specify the equivalent of formatting for screen reading devices. Stylesheets can be altered or substituted without and need to alter the markup of the document itself.

2.1.3 Web-based Document Technology

XML (eXtensible Markup Language) is the basis for XHTML and a host of other languages that have been proposed, if not actually used, for marking up different types of content on the Web. XML is the format used to deliver RSS feeds and podcasts. Almost all the formats and languages proposed for implementing the W3C's "Semantic Web" are based on XML, In particular, XML can be used as a concrete syntax for RDF (Resource Description Framework), which is intended to provided a standard for metadata on the Web. It is also used as a format for exchanging data between web applications, and has found many applications off the Web. XHTML is used to define ODF (OpenDocument file format) and Microsoft's

OOXML (Office Open XML), both of which are used as formats for office documents, and provides the syntax for Apple's property list files, used to record preferences on Mac OS X.Adobe used XML to define MXML, the layout language for Flex, and XFL, for exchanging Flash documents.

At its simplest, XML can be used as a markup language, like XHTML, to apply tags to documents so as to identify their structural elements. Unlike XHTML, thought, XML does not provide a fixed set of elements. In effect, it lets you make up your own. For simple purposes, such as exchanging data between a blogging application and a desktop blog editor, this is adequate, but for more important tasks it is helpful to be able to impose constraints on which elements may be used, what attributes they may have and which elements may be contained in others. This allows a program that processes the XML data to verify that it is correctly formed and includes everything it should and nothing it shouldn't.

A formal definition of a set of elements and their attributes, together with constraints on the way they may be combined, can be created in the form of an XML Document Type Definition (DTD). In effect, a DTD defines a specialized markup language. XHTML is defined by a DTD, for instance. You can therefore consider XML not just as a markup language, but also as a markup metalanguage (a language for defining markup language).

Languages defined by XML are often called XML-based languages. They all share the basic notation for writing tags and entities that we described for XHTML,

but have different sets of elements and their own rules about how these elements can be used. The common syntax makes it feasible to mix elements from several XML-based languages in the same document.

XML 1.0 was adopted as a Would Wide Web Consortium Recommendation early in 1998. It was intended to provide a new foundation for the Web, which would be built by mixing XML-based languages including XHTML, SVG, MathML (Math Markup Language) and SMIL (Synchronized Multimedia Intergration Language). Matadata would be added using RDF and the semantics would be described for processing by machines using OWL (Web Ontology Language). Improved linking would be provided by XLink and better forms by XForms. XSL (the eXtensible Stylesheet Language) would allow radical restructuring of documents to be performed, which in turn would allow more extensive control over appearance than CSS provides. XML would even bootstrap itself: DTDs would be replaced by "schemas" which would define XML-based languages using XML syntax.

Things haven't worked out that way, though. The XML-based Web has been met by a mixture of indifference and incomprehension by the majority of the Web design community, and by positive hostility from the Web browser makers. The leading proposal for a successor to XHTML 1.0 is HTML 5, which can be "serialized" as XML, but can also be serialized using a custom syntax based on HTML 4, which is recommended instead. XForms has attracted little attention and HTML 5 includes an alternative extension to the existing form elements. The metadata formats considered essential to the Semantic Web are only used in small

specialized communities. The XML-based languages specifically concerned with the subject matter of this book are being little used. SVG is finally being implemented, at least partially, in most browsers, but continues to be overshadowed by Flash as a vector format. SMIL's only popular application to date has been the embedding of advertisements in media streams.

Despite all this, XML is worth knowing about. Certain features of XHTML only make sense if you understand the relationship between XHTML and XML. Some XML-based languages for Web metadata are likely to become increasingly important, even if others continue to be seen as irrelevant. XML's role in Web services, allowing data to be exchanged between Web applications, is firmly established. Although the dramatic change to a Better Web Built on XML may not have occurred as foreseen, XML's other uses continue to grow. Furthermore, because XML is easy to read, looking at an XML-based language, such as SVG, can provide insight into the way media are represented.

In Thailand, The only standard for Thai character set for information interchange is TIS 620-2533 (1990) which is defined by the Thai Industrial Standard Institute (TISI), Ministry of Industry. It defines two eight-bit character code sets by extending ISO 646-1983 and IBM GX20-1850-4 (EBCDIC), respectively. TIS 620-2533 (1990) is indeed a minor correction of the previous one, TIS 620-2529 (1986). The character set table remains the same. Only some detailed descriptions are added for compliance with international standards. The most Thai documents are encoded using the extension of ISO 646 in TIS 620. In the internet, there has been a confusion of

Multipurpose Internet Mail Extensions (MIME) character sets for Thai e-mails and web pages among the ad hoc solutions, such as using iso-8859-1, x-user-defined and windows-874, until, in 1998 tis-620 character set registration with the Internet Assigned Number Authority (IANA), according to RFC 2278, for information interchange on the internet. IANA registered it in the same year. Therefore, the implementations of web-based system that process Thai document have limitation in document format. Because there is only one standard for Thai character set for information interchange which is TIS – 620.

2.2 Metadata

This section presents the concept and significance of metadata usage in organizations. Nowadays, knowledge management in organizations has become a vital challenge. There are knowledge is placed in huge computer system in the form of digital documents. Digital documents can be based on individual stand-alone document files such as Adobe PDF, MS Word and MS PowerPoint documents or on internal document types in the computer system. Usage of digital documents has introduced many new sharing and efficiency dissemination knowledge. However, usage of computer systems can easily limit knowledge sharing while the correct documents are difficult to locate. With a fast increasing collection of documents, locating the correct document becomes more challenging.

Metadata can be used to give descriptions of the document. These descriptions can be a part of the data used for document querying and retrieval. This is allowing

new users to gain knowledge of the existing recourses and their most central characteristics. The simple definition of metadata is “data about data” (Dublin Core 2009). This is not an informative definition. Therefore, a number of more informative definitions have been developed (Dublin Core 2009). According to W3C (2009), metadata, structured data about data, improves discovery of and access to such information. The effective use of metadata among applications, however, requires common conventions about semantics, syntax, and structure. Individual resource description communities define the semantics, or meaning, of metadata that address their particular needs. Syntax, the systematic arrangement of data elements for machine-processing, facilitates the exchange and use of metadata among multiple applications. Structure can be thought of as a formal constraint on the syntax for the consistent representation of semantics.

These metadata are based on a pre-determined and standardized metadata schema which present the possible description elements and the valid content of these elements. The metadata descriptions can be a part of the data for document querying and retrieval by presenting the recourse and its most central characteristics in query results. A major challenge is to create metadata descriptions due to user knowledge requirements, timely metadata registration processes, human costs and the ongoing challenge of more documents being published. These issues can be reduced or even avoided entirely by enabling computer software to generate metadata instead of a supplement to manual metadata actions.

2.2.1 The Nature of Metadata

Metadata describes different qualities of the content essence such as format, semantics, or status. The division between content essence and metadata is not always clear and depends on the context; what may be metadata for one purpose might be considered as part of the content essence for another.

Two important agreements of metadata must be reached before it can be used in the content value chain. Firstly, there must exist an agreed format for metadata, the grammar, and methods to make different formats of metadata compatible so that organizations can access and use metadata in their activities. Secondly, the semantic interpretation of the metadata forming the vocabulary must be agreed upon, so that content can be processed intelligently (Curtis et al. 1999).

Applications and users have different needs that must be reflected in the metadata. For example, if the metadata is used for producing advanced content-based products, metadata typically describes qualities related to semantics, authoring, formatting, status, and intellectual property rights of the content. The following Table 2.1 describes metadata fields used in the Dublin Core standard (Dublin Core 2009), which was developed by the library community for resource discovery on the World Wide Web. Although Dublin Core is at the time of writing one of the most widely accepted and adopted standards for describing content essence, its usefulness and expressiveness, the capability to express different semantic aspects of the content essence, is highly limited due to its simplicity and generality. Many of its fields do

not have agreed vocabulary leaving a lot of room for different interpretations of the meaning of the field. For example, the “subject” field does not have any further structure and merely reserves that field for the “topic of” the content (Dublin Core 2009).

Table 2.1 Detail of element set (Dublin Core 2009).

Field	Description
Title	A name given to the resource.
Creator	An entity primarily responsible for making the content essence of the resource.
Subject	The topic of the content essence of the resource.
Description	An account of the content essence of the resource.
Publisher	An entity responsible for making the resource available.
Contributor	An entity responsible for making contributions to the content essence of the resource.
Date	A date associated with an event in the life cycle of the resource.
Type	The nature or genre of the content essence of the resource.
Format	The physical or digital manifestation of the resource.
Identifier	An unambiguous reference to the resource within a given context.
Source	A reference to a resource from which the present resource is derived.
Language	A language of the intellectual content essence of the resource.
Relation	A reference to a related resource.
Coverage	The extent or scope of the content essence of the resource.
Rights	Information about rights held in and over the resource.

Different kinds of metadata refer to the content essence at varying levels of granularity. Some kinds of metadata describe overall qualities of the content essence (e.g. content length), while others describe just certain parts of the content essence

(e.g. opening paragraph keywords). Metadata can be kept with its content essence (tightly-coupled or implicit metadata) or kept elsewhere (loosely-coupled or explicit metadata). The latter can be stored and transmitted separately, whereas implicit metadata is stored and transmitted together with the content essence.

Metadata is essential if the content essence cannot be used without its metadata. An example of essential metadata is information related to the compression or decryption of content essence.

Metadata can be dynamic or static, depending on its usage and the nature of the content domain. Static metadata remains unmodified from the creation to the last time it is used. An example of static metadata is author information. Dynamic metadata changes over time and requires periodic refreshing or recreation, such as updating the number of available stories in an online news portal. Temporary metadata is created only for a certain phase in the content value chain so that it may be intentionally destroyed after it has served its useful purpose. Examples of temporary metadata include status and workflow scheduling information.

Not all metadata is public. Private metadata is well defined, but its detailed description is not publicly available, although it could be obtained, for example, through licensing. Private metadata might also have been publicly defined but not yet been accepted as an identified kind of metadata by other participants of the content value chain.

The participants in the content value chain should preserve and distribute all such metadata that may have use for some other partner in the content value chain, even though the participant itself does not anymore need the metadata. For example, after creating a web page using compressed content essence, there is no need for preserving metadata related to compression, but if the compressed content essence is available to other participants in the content value chain, that metadata should be passed to other participants as well. Likewise, if a media company does not use or recognize a certain kind of special metadata incorporated in the incoming content, that special metadata may still be important for some other participant in the content value chain, and as such should not be discarded from the output.

In some cases the metadata is not needed for the actual deliverable, but it can be used to produce advanced by-products. An example of this can be seen in automobile manufacturing. Even though the physical product, a car, can hardly be categorized as an advanced content-based product, its content essence and metadata can offer new business possibilities, and used, for example, on the manufacturer's web site to promote the car.

2.2.2 Categorization of Metadata

As described in the previous section, metadata has a variety of characteristics leading to categorization from different points of view. For example, categorization can be based on the usage of metadata, stages in the life cycle of metadata such as creation, usage, and maintenance, or by the characteristics of metadata. If we use

role-based categorization as suggested by (Boll et al. 1998) for classifying different kinds of metadata, a possible division could be structural, control, and descriptive metadata.

Structural metadata describes the structural characteristics, the format, of the content essence, but does not contain information about what the content essence actually means. Structural metadata has therefore no relation to previously discussed structured content, which emphasizes the co-existence of metadata and content essence. Examples of structural metadata include decoding information such as video, audio, and graphics formats, compression data, composition and synchronization information, as well as information on sequencing the content essence. Structural metadata is often tightly-coupled with the content essence and is essential for its usage. Different media platforms and advanced content-based products have their own requirements on structural metadata. These media platform and product specific requirements include, for example, the support for audio and video using formats like QuickTime.

Control metadata is often created and used for controlling the flow of content in the content value chain. Control metadata is used to determine whether content is ready for the next step in the content value chain. Control metadata is quite often temporary in nature as opposed to more permanent semantic metadata. Some examples of control metadata are machine control, quality of service, and error management.

Descriptive metadata can be divided into two subcategories, contextual metadata and content-based semantic metadata. Contextual metadata describes the environment and conditions of content essence and its creation. This includes geospatial information, timing information, as well as information on the equipment used to produce the content essence. Semantic metadata describes semantic qualities of the content essence answering the question what the content essence means. Semantic metadata is needed for the processing or usage of the content essence. It describes such qualities as the subject, location, names, and style of the content essence. The keywords of a news story are an example of semantic metadata. Semantic metadata is typically used in advanced content-based products, such as in a personalized news service, where the metadata is used to determine whether the user might be interested in the content essence or not. Semantic metadata requires an agreed semantic interpretation before it can be used by different participants of the content value chain. If the content provider does not produce semantic metadata according to agreed standards, or if a common agreement for interpreting semantic metadata is missing, computer-based processing that relies on the semantics of the content essence is likely to fail, or the quality of the outcome is likely to suffer.

Descriptive metadata can also contain information about how the content essence can or should be used and is thus closely related to control metadata. Examples of this kind of usage information are intellectual property and access rights, as well as information on supported media platforms.

2.2.3 Metadata Standardization

This chapter contains a brief overview of some standards related to semantic metadata. As these standards are emerging, evolving, and changing rapidly making the comparison difficult and the results quickly outdated, I have excluded detailed and exhaustive analyze from my work and present just few of the standards that are related to news content and relevant to this work. Comments on these standards are partially my own observations, partially derived from references and other materials discussing standards on news content such as (Dumbill 2000).

The standards presented in the Table 2.2 have varying degrees of ambitiousness and are at varying levels of completeness and implementation. They also more or less overlap each other. Development in the standardization of metadata is fast, and changes take place frequently, so it is important to check the validity of presented information from the references. Readers interested in more detailed information on the standardization of metadata should see for example (Saarela, 1999), which contains detailed information on XML and associated standards.

Standard Generalized Markup Language (SGML) and Extensible Markup Language (XML) form the basis for most metadata standardization efforts. They define basic methods for using tags for marking semantic metadata in electronic documents, but leave the definition of those tags to the users. Although the two core standards, SGML and XML, are relatively mature and established, standards related to semantic metadata are still insufficient to describe content essence in greater detail,

even though some alternatives, such as NewsML11 and industry specific efforts 12, have lately focused their efforts towards this goal.

The creation of semantic metadata related standards is a challenging task and requires participation and agreement of impacted participants in the content value chain. The lack of sufficient standards has motivated media companies to develop their own proprietary standards, or to enhance the existing standards with proprietary extensions. However, this development is likely to be a temporary step on the way to more shared and open standards. For example, the Reuters news agency has developed its own proprietary standard for describing content. If a content provider does not adhere to the standard, its content is excluded from distribution. Consequently, Reuters has lately been shifting away from proprietary solutions and into more open alternatives, such as NewsML (Reuters, 2000).

Table 2.2 Some metadata standardization.

Standard	**Purpose and brief description**
	Advantages
	Disadvantages
Dublin Core	Dublin Core (DC) is a standard for metadata developed by the library community for resource discovery on the World Wide Web (WWW). It captures both semantic metadata and contextual metadata about the content essence. Dublin Core was designed to enable searching of documents across heterogeneous databases and schemas by capturing bibliographic and other descriptive information of content essence. It provides a simplified set of 15 metadata fields that form the core ontology of the standard.
	Dublin Core is widely used for archival purposes in libraries. Dublin Core has international consensus as a mature and robust standard. Dublin Core is simple and open standard that has been used as a core in other standardization efforts.
	Dublin Core has limited support for semantic metadata. Dublin Core requires additional agreements before it is suitable for describing semantic metadata that can be used automatically by computers. Dublin Core extension beyond the agreed set of metadata fields requires changes in the standard.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
ICE	Information and Content Exchange (ICE) is an XML-based standard defining a vocabulary and protocol for the delivery of content and the management of relationships in syndication. The ICE protocol defines the roles and responsibilities of media companies and their customers, defines the format and method of the delivery of content, and provides support for management and control of relationships in the content value chain.
	ICE has some commercial implementations and relatively large support among media companies and their customers. ICE can be managed an important part of the content value chain, namely the delivery of content and relationships between media companies and their customers.
	ICE is not developed by standards body W3C, which in turn weakens its acceptance outside of the originating companies. The public version of ICE does not have support for describing semantic metadata. The future of ICE is unknown; it will most likely be merged with other standards.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
NewsML	NewsML is an XML-based standard for combining, representing and managing news content irrespective of its media platform, format, or encoding. NewsML is developed by the International Press Telecommunications Council (IPTC). NewsML extends beyond newswire providers and newspapers and focuses on a wider variety of participants in the content value chain. The first version of the NewsML, NewsML v1.0, was released to public in October 2000.
	A number of large companies and organizations have participated in the standardization effort. Semantic metadata in NewsML is based on the widely accepted NITF standard (although it can utilize other standards as well). NewsML is independent of media platform.
	NewsML contents suffer from overlapping and compatibility issues, especially in relation to semantic metadata, with other standards such as PRISM. NewsML is not yet in widespread use and implementations are rare.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
NITF	News Industry Text Format (NITF) is a widely used XML-based standard for marking up the semantic metadata of news content. NITF was developed in co-operation between two major standards organizations in the news industry, the International Press Telecommunications Council (IPTC) and the Newspaper Association of America. The following characteristics are covered in the newest version of NITF, version 3.0: Who the news is about, who owns the copyright, and who may republish it. What subjects, organizations, and events the news covers. When the news was reported, issued, and revised. Where the news was written, where the action took place, and where it may be released. Why the news is newsworthy.
	Powerful organizations and companies are involved in the standardization effort. NITF is used as a basis in other standards such as NewsML.
	NITF support for other than news-based media products is unclear. NITF does not contain methods for defining ontology.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
PRISM	Publishing Requirements for Industry Standard Metadata (PRISM) is an XML-based standard for describing metadata, which is needed in syndicating, aggregating, post-processing and reusing content in magazines, news, catalogs, books, and journals. PRISM aims to provide standardized vocabularies for metadata in order to enable the interoperability of all kinds of content. The PRISM authoring group released version 1.0 of PRISM to the public in April 2001.
	PRISM has a wide coverage over a multitude of characteristics required in the content value chain. PRISM support for defining ontologies.
	PRISM is not in widespread use due to its novelty. Acceptance outside of originating companies might be limited due to the closed nature of how the standard was developed. Compatibility and overlapping with other existing standards might cause issues. PRISM supports only a subset of RDF. Additionally, PRISM seems to overlap with NewsML. Even though the PRISM group is working on compatibility with NewsML, the outcome is still unclear.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
RDF	Resource Description Framework (RDF) (Lassila and Swick 1999) is an XML-based standard providing a common syntax to describe metadata about a resource such as a document on the World Wide Web. RDF is a recommendation from the World Wide Web Consortium (W3C) and is used as the core mechanism for Semantic Web at the W3C enabling the exchange of knowledge on the World Wide Web.
	RDF allows the use of multiple metadata standards through the XML namespace mechanism. This makes RDF flexible and extensible for future needs. RDF can be added structure to content and improves the computerized use of metadata as long as the semantics of the ontology are shared. RDF has potential to become the default structure for describing semantic metadata.
	Simple metadata without complex relations can be described without RDF questioning the need to learn or use RDF. RDF specific tool support is still limited. RDF syntax is moderately complex, at least for nonprogrammers.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
RSS	RDF/Rich Site Summary (RSS) is a simple format for metadata related to online news. RSS has evolved through different versions without backward compatibility such that versions 0.91 and 1.0 are in some cases considered as different standards. Version 0.91 is inspired by RDF, but not strictly conformant to it. Current version, RSS 1.0, is compatible with RDF and has roughly the same descriptive abilities as PRISM and NewsML.
	RSS is a lightweight and flexible standard that is easy to understand and simple to use without large investments in resources.
	Compatibility problems between different versions. The future of RSS is unknown. Netscape, the main advocate for RSS, dropped support for RSS recently, and its role as the host for the standard has not yet been filled.

Table 2.2 (continued) Some metadata standardization.

Standard	Purpose and brief description
	Advantages
	Disadvantages
XML News	XMLNews is an XML-based standard for describing the semantics and context of content essence. XMLNews consists of two parts: 1.) XMLNews-story, designed as a subset of the older 1998 version of NITF, describing the semantics of content; and 2.) XMLNews-Meta, an RDF-based extensible vocabulary describing news resources.
	XMLNews are used in commercial applications and newsfeeds from companies such as iSyndicate22. XMLNews is simple.
	Availability and quality of semantic metadata varies in implementations. Some sources use only some of the possible tags, and/or their semantic metadata has low quality. XMLNews contains only basic support for semantic metadata, which is then extended with nonstandardized, source-specific semantic metadata. Standard is outdated and not necessarily compatible with other standards. XMLNews is likely not to be developed further as other standards are taking its place.

2.2.4 Summary

In summary, among the current metadata standards, Dublin Core has the potential of being adapted as an international standard for resource description and discovery on the web because of its simplicity. Moreover Dublin Core has received widespread acceptance amongst the resource discovery community and has become unofficial the Internet metadata standard.

The collection of the metadata elements within describes a document is known as a metadata element set (Dublin Core 2009). Some of the common characteristics of all metadata include semantics, syntax and structures. Semantics refer to type and content of metadata elements, syntax refer to the way in which content is structured according to a specify grammar. Metadata standard may range from simple syntax like Dublin core metadata element set to complex coding systems such as mark up languages like SGML. Structures refer to the over all architecture that contain metadata content and syntax. Metadata may be embedded in the digital object and extracted as needed or they may reserve in separately indexed databases. The data can be contained in a variety of architectural structures including Z39.50 compliant library catalogues, proprietary databases or Resource Description Frame work (RDF) standard.

Therefore, the appropriate metadata standard for this research is Dublin Core because the basis of Dublin Core is for document discovery on the World Wide Web. In Thailand, the review also indicated that Dublin Core metadata is in used for

describing electronic resources for Thai government publications collections, Thai research databases, and Thai e-theses and dissertations. Several projects have been developed and implemented in government agencies, special libraries, private sector and academic libraries. Government digital databases are growing rapidly through collaboration between government agencies under the National Science and Technology Development Agency (NSTDA) and the co-operation of academic libraries. Then Dublin Core is a familiar standard for collaboration between Thai librarians and Thai domain experts.

2.3 Approaches for Metadata Generation

The automatic metadata generation algorithms are constructed to take advantage of one or more available data sources. The algorithms are constructed based on rules enabling gain access to the data source, identify desired content, and collect this information and storing them in accordance with a metadata schema. The automatic metadata generation algorithms which is used existing metadata is referred to as harvesting algorithms while algorithms that create new metadata are referred to as extraction algorithms. This section presents the methods for generating document metadata which is composed of harvesting, extraction based on visual characteristics, extraction based on natural language and extraction based on the document code.

2.3.1 Harvesting

The approach of harvesting the existing document metadata can be regarded as the easiest way to generate document metadata. A number of commonly used document types store descriptions of the file content as part of the file in their documents' code. These metadata can be created by content creation software, by users or both. There are two main reasons for including harvestable metadata: (1) For allowing content creator software to correctly identify the document type and enable encoding and interpretation of the document content in the intended way. E.g. there are currently eight versions of the Adobe PDF file format where distinction between versions is based on version metadata. (2) To enable more usability for the document creator. These harvestable metadata are therefore also commonly displayed in different user interfaces to enable the user to more easily locate the desired document. E.g. the song name, album, release date and artist name is frequently displayed for MP3 sound files.

Specific file types can contain extensive harvestable metadata descriptions. E.g. the MS Office file types include logistical metadata regarding the creation, last saved and last printed dates, semantic metadata with the name of the user who performed the previously listed actions, title, keywords, description and technical elements regarding the number of characters, words, pages and slides which the document consists of. JPEG images can contain an XML-based section (EXIF) which can contain data regarding camera settings when a picture was taken, geographic location (GPS coordinates) and technical descriptions of the image (resolution (dpi),

dimensions (horizontal and vertical number of pixels) etc.). Adobe PDF documents can contain multiple metadata sections, allowing metadata based on multiple metadata schemas to be included in an individual file. An extensive range of elements is supported used.

Different file formats have different approaches regarding where in the physical file where the metadata is stored, how these data are coded and their used metadata schema. Gaining access to the harvestable metadata therefore require knowledge of the structure and interpretation of the specific version of the file format. There is therefore no general method of gaining access to harvestable metadata. Projects using harvestable metadata are therefore concentrated on specific file formats. The most common file format to study is HTML due to its frequent usage on the Internet and since it uses a text-based document code format. This allows easier to gain access to the metadata and other file content than by working with binary file formats, such as PDF and Word.

2.3.2 Extraction Based On Visual Characteristics

Metadata harvesting is limited to the specific elements which are present in the document. Content creation software (user applications) is known to systematically generate false metadata. This is a reason for why many document projects do not use this data source. As a consequence, many projects enforce extraction of metadata rather than harvesting. This approach uses a content presentation application to generate a visual representation of the document. Such applications can attempt to

present the document as if it were presented in its native content creation software or as a print-out. This representation is created based on the document formatting and the intellectual content created by the document user(s). The visual representation is used as data source for rules adapted to identify and extract specific visible document content.

The algorithms based on visual characteristics use the visual appearance of the document to identify document content. There is therefore an extensive demand for human efforts in generating the rules, determining rule weights and adapting the rules to work together in generating the desired results. This is further complicated if the document types are evolving, e.g. if a new content creation software version uses the file format in a new ways. Then the AMG algorithms need to be updated to tackle documents created using both the old and new software. As a consequence, rules which were correct earlier can become false or require a re-shuffle of the labyrinth of rules to determine the best candidate entity.

2.3.3 Extraction Based On Natural Language

An alternative to the rules based on visual characteristics has been developed in the form of natural language rules. This approach also uses special content presentation applications to retrieve only the intellectual content of the document, creating a plain text data source in which the rules based on natural language is executed. Such algorithms commonly include collection of unique words and comparisons of the document vocabulary against reference ontology for keyword

generation placed in the Document publishing system's context information. Natural language based algorithms can function by comparing content from different sections of the document against each other and by weighting the value of specific words and phrases. The natural language approach requires extensive local knowledge to adapt the algorithms to the way the local users are using their language and the vocabulary used. The algorithms need to handle different forms of words, synonym words and synonym phrases without confusing or mixing documents. To cope with this, technologies such as thesauri and ontology are frequently used. However, these technologies are manual labor intensive to generate and maintain. It requires extensive knowledge of how the language is used. This makes the developed vocabularies case- or subject specific making their general usage limited. This limits the usage of such technologies to the specific subjects and local contexts in which they were developed. It is therefore a solution which has been experienced adapted on subjectspecific document collections. Usage of rules based on the natural language approach is most commonly used to generate entities for more general elements, such as summaries, descriptions and keywords. Though, this method has also been used to generate titles.

There is an extensive demand for human documents in generating the rules, determining weights and adapting the rules to work together in generating metadata. This is further complicated if the document types and subjects are evolving. Then the AMG algorithms need to be updated to tackle new and old challenges. As a consequence, rules which were correct earlier can become false or require a re-shuffle of the labyrinth of rules to determine the best candidate entity. In addition it is

becoming more normal with multilinguistic user environments, which further complicates for natural language based algorithms.

2.3.4 Extraction Based On Document Code

Extraction based on the document code uses the document code directly without the need of additional content presentation applications to interpret the document content in order to create a usable dataset for the metadata generation efforts. This enables full access to all the content of the document code without potential contamination from content presentation applications due to their interpretation of the document code. Basing metadata generation efforts directly on the document code avoids many of the challenges which face extraction algorithms which are based the visual presentation of the document. Using the documents' code allow the metadata generation algorithms to gain direct access to the user specified document content. This avoids having to use technologies such as Optical Character Recognition (OCR) or other converting applications to gain access to the document content and its formatting. This is regardless of the visual presentation and the language of the intellectual content used within the document.

The document code of the document can be used to gain document descriptions which are automatically generated, though not presented as metadata. E.g. the language used within the document is automatically included in MS Office documents to enable use of spell-checkers. The MS Office application practices automatic labeling of text-based content sections or even single words. It is therefore possible to

distinguish between the specific content sections and their used language within this section. This would be a valuable tool in which to base natural language algorithms upon, since it can exclude content in languages not covered by the natural language algorithm. This can avoid one of the major challenges in introducing language based algorithms in a multi-linguistic user environment, such as a university. The potential of using this approach has been limited by the understanding of the document codes. This situation is currently changing as commonly used document file types are being moved towards into non-binary, standardized file formats based on XML code. Such formats have been introduced by Microsoft (MS) for their MS Office document formats (MS Word, MS PowerPoint and MS Excel). Their new file format is based on the Open XML standard. MS supplies a lossless converter application between the "old" binary and the "new" XMLbased file formats. This enables full insight into the document code of these files formats. This in turn opens for a range of AMG efforts based directly on all the content of the document code without document content distortion.

Usage of the document code require extensive knowledge of how applications use file formats in order to avoid data sources which are used to present content which do not reflect upon the document. This includes new usage patterns enforced by new applications or application versions.

2.3.5 Summary

Each one of the metadata extraction algorithm approaches has their own strength and weaknesses.

Harvesting uses the easy to access and collect entities from existing metadata stored as part of the files' document code. This approach's main weaknesses are: (1) The limited amount of elements in practical use. (2) Uncertainty regarding if the used elements contain entities which reflects upon the document. (3) Few people are aware of existing metadata and hence few people place efforts in generating and correction.

Extraction based on visual characteristics approach can be used to identify and collect a large number of elements. It intends to collect content which the user has specified. This approach's main weaknesses are: (1) Its requirements regarding knowledge of the used documents. (2) Requires standardized formatted documents. (3) Can require a labyrinth of rules which need to work together. (4) Issues regarding multiple, candidate elements.

Extraction based on natural language approach has potential of generating semantic metadata; metadata which even humans can find difficult to generate such as classification, subject, keywords and description. Though it require: (1) Extensive knowledge of the document contexts, limiting it to specific subjects and specific languages. (2) Limited to specific elements, requiring it to be used in companion with

other metadata efforts for practical usage. (3) Do not scale to a general purpose context or a multi-linguistic environment.

Extraction based on the files' document code approach can be used to collect all user specified content from template sections regardless of visual document presentation or language of the intellectual content. It enables blank metadata generation results if no section content is collectable, avoiding generating of multiple candidate entities. It can be used to collect document descriptions which are part of the document code, such as references, language tags, illustrations and tables. It can provide extensive descriptions of the document usable to increase the correctness of other metadata generation algorithms. This by providing a data source based on facts rather than software based judgment and by providing direct access to the main document content. Using this

Therefore, the appropriate metadata generations approach for Thai document is based on natural language approach. Because the structure and characteristic of written Thai is highly ambiguous, which requires more sophisticated techniques than are necessary to perform comparable a task in most European language, and large amounts of domain knowledge to cope with these ambiguity.

2.4 Information Extraction

This section discusses theoretical background on Information Extraction (IE). IE is one of the Natural Language Processing (NLP) tasks. The purpose of Information Extraction, in terms of the NLP domain, is to create a system that process the digital documents written in any of the natural languages and identify relevant information. In other words such system will be able to discover semantic information in the documents. The Information Extraction System can be also perceived as a set of linguistic tools and resources that combined are used to performed specific task related to the natural language processing. In general the IE System can be presented as a black box (see figure 2.1) that uses linguistic resources to fulfill given task. A user delivers documents in a natural language to the IE system and receives the result of the processing. The results depend on the task characteristic.

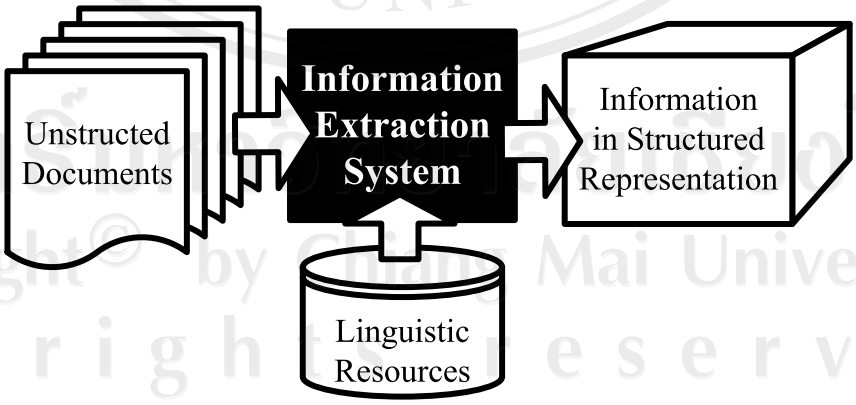

Figure 2.1 Information Extraction system as a black box.

2.4.1 Information Extraction System Architecture

The general architecture of IE systems is based on a pipeline processing where set of modules is executed in given order. An output of one module is an input for another. The order in which the subsequent modules are executed is essential because some modules provide or require information that is required or provided by other modules. Depending on the task specification and the language characteristic different types of modules are plugged into the pipeline. Figure 2.2 presents basic modules that are used in Information Extraction Systems.

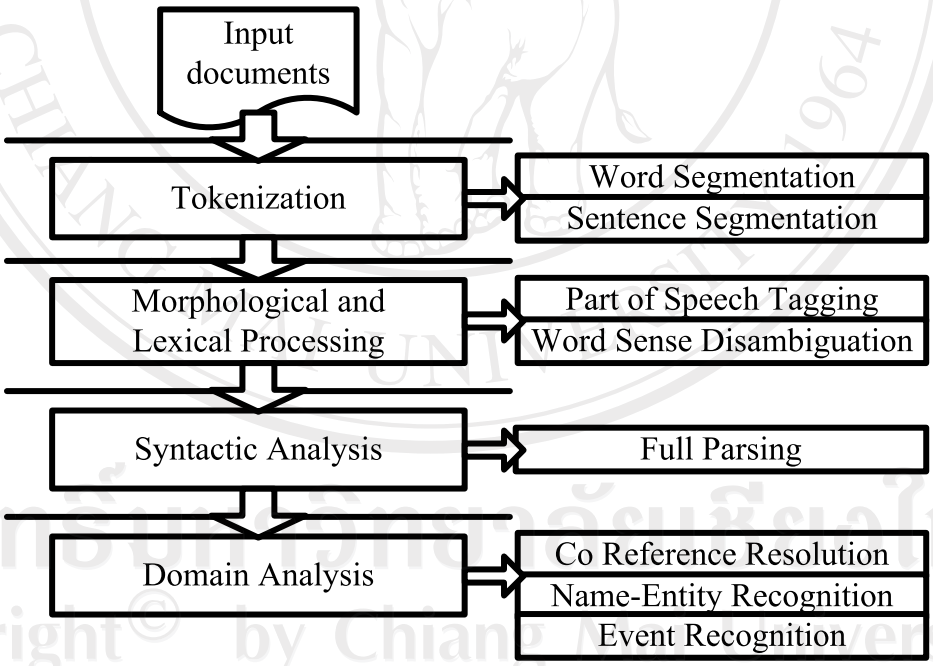

Figure 2.2 Basic modules in Information Extraction system.

Tokenization module includes word segmentation and sentence segmentation. Word segmentation is a process of dividing text into words. This problem is trivial

for languages that have explicit word boundaries like white spaces. However in some languages (for examples Chinese, Thai and Japanese) there is no explicit boundary and to solve this problem more complex techniques must be applied. Sentence segmentation is a process of dividing text into sentences.

Morphological and lexical processing module includes part of speech tagging and word sense disambiguation. Part of Speech Tagging (Allen 1995) is a process of determining word part of speech, based on definition and the context in which the word appear (like relation to other words in the sentence). In addition the POS tagging identifies the case, the number and the person for every token. It is common that after POS tagging some words get more than one interpretation. For example word play can be used as verb and represent an action of playing a game or on a musical instrument, noun and represent a piece of writing performed in a theater. Word Sense Disambiguation (WSD) is the process to determine which sense (meaning) of a word is activated by the use of the word in a particular context, a process which appears to be largely unconscious in people.

Syntactic analysis module includes full parsing technique. Full Parsing is a process of transformation the sentence into a tree structure that depicts relations between tokens (Allen 1995). In such structure node children represent arguments of the predicate in the node.

Domain analysis module includes co-reference resolution, name-entities recognition and event recognition. Co reference resolution (Allen 1995) is the

process of identification different words (in general) in a document that refers to the same instance. For example, in a fragment of a document “This is Mark. He is my best friend.” words Mark and He refer to the same person. Named Entity Recognition/Identification (Baeza-Yates and Ribeiro-Neto 2002) also known as Named-entity Recognition, is a process that leads to locate and classify atomic elements in the text into predefined categories like people, companies, locations, expressions of time, monetary values and many others. During this process some semantic information is attached to the document. Event Recognition (Baeza-Yates and Ribeiro-Neto 2002) is a process similar to Named-entity identification, however, the information attached to elements of the text is more precise and takes into consideration not only semantic class but also meaning of the element. For example if we have a document that treats about a concert announcement and it contains two dates: one represents the concert date and the other the date up till tickets can be bought. In this case both dates belong to the same semantic category but have different meanings.

2.4.2 Measuring Information Extraction System Performance

The common way used to compare the performance of different Information Extraction systems is to compare the ratio of correctly and incorrectly classified positive instances (Baeza-Yates and Ribeiro-Neto 2002). The negative instances are not taken into consideration in the evaluation because the number of negative instances is very prevalent. Also the accuracy of classifying the positive instances is much more important for Information Extraction then the classi fication of the

negative instances. The results of the classification task are divided into four groups which are used to evaluate the system performance (Baeza-Yates and Ribeiro-Neto 2002). They are as follows:

True Positives (TP): positive instances correctly classified as positive.

True Negative (TN): negative instances correctly classified as negative.

False Positives (FP): negative instances incorrectly classified as positive.

False Negative (FN): positive instances incorrectly classified as negative.

There are two metrics that represent the system performance with respect to the number of TP. It is Precision (P) and Recall (R). In addition third metric namely F-measure is used that combines the value of the Precision and the Recall. Below the equations of the Precision, the Recall and the F-measure are presented:

Precision defined as:

$$P = \frac{\text{number of relevant instance in the result}}{\text{number of instance in the result}} = \frac{TP}{TP + FP}$$

Recall defined as:

$$R = \frac{\text{number of relevant instance in the result}}{\text{number of all relevant instance in the set}} = \frac{TP}{TP + FN}$$

F-measure is a combination of precision and recall and is defined as:

$$F = \frac{(\beta^2 + 1) * P * R}{\beta^2 * (P + R)}$$

β is a parameter that represents relative importance of P and R.

The common way to present the tests result for different configuration of learning and testing parameters is a plot of precision versus recall. Figure 2.3 shows an example plot where each point on the plot represents single test case with some configuration of parameters. This type of plot depicts the relation between recall and precision - the higher recall is the lower precision. This relation can be approximated to the curve presented on the figure 2.4.

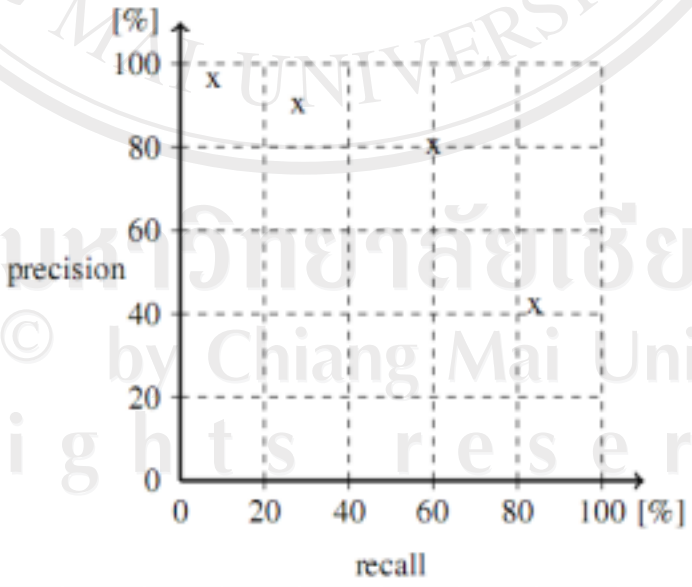

Figure 2.3 Example of precision versus recall plot.

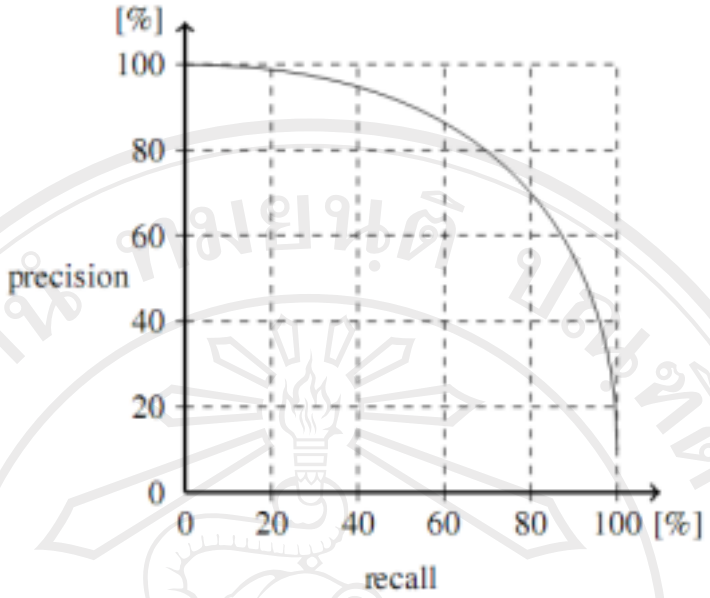

Figure 2.4 Example of precision versus recall plot approximated to curve.

2.4.3 Cross Validation

When a system is using rules learned with any machine learning method or technique instead of rules created manually there is a need to take into consideration one more factor that is related to the problem of estimating generalization error. In this case the whole set of instances cannot be used in the learning process because there is no point in testing the system on the same set. Such result will be misleading because the systems performance will be based on testing the "known" instances. More important is how the system will cope with the unknown instances. One of the common methods in this case is Cross-Validation (CV) (Kohavi 1995). The general idea of CV is to divide the set of instances into several subsets and test the system using different combination of the subsets as the training and the testing sets. For example in 10-fold Cross Validation the set of instances is divided into 10 subsets and

in each test 9 subsets are used as the training set and the 10^{th} as testing set. In every test different subset from the 10 is used in the testing.

Metrics presented above are widely used in comparing Information Extraction systems performance. However those values cannot be directly compared between different systems unless the tests were conducted on the same set of instances. Moreover the top score is not always 100%. It is caused by the fact that natural language is not always unambiguous and can be interpreted in different ways by different people. So there is a need to estimate the top score by tests conducted by an independent group of people. In other words the system cannot learn to recognize relevant information if group of people is not coherent with the interpretation. However the process of estimating the top score is very time-consuming and requires additional resources.

2.4.4 Summary

In summary, Information Extraction (IE) is a type of information retrieval which goal is to automatically extract structured information, i.e. categorized, contextually and semantically well-defined data from a certain domain, from unstructured machine-readable documents. The significance of IE is determined by the growing of information available in unstructured form (i.e. without metadata), for instance on the Internet. This knowledge can be made more accessible by means of transformation into relational form or by marking-up with XML tags.

Therefore, the basic architecture of IE consists of various natural language processing components that perform an analysis on the Thai document. First, the morphological level of linguistic processing is concerned with the processing of recognizable paragraphs and the word forms. The morphological analysis is considered important in IE for Thai, because there is no word boundary in Thai language. The effectiveness of term searching relies excessively on the performance of the segmentation process. Next, the technique of deal Thai lexicon with the analysis of Thai syntactic features to resolve part-of-speech is applied. Finally, a phrase structure grammar can be used to produce a tree structure for Thai sentence.

2.5 Case-based Reasoning

This section discusses theoretical background on CBR. The research on CBR can be traced back to the work of Roger Schank's dynamic memory and Memory Organization Packet (MOP) (Schank 1982). According to Schank (1982), "a dynamic memory is one that can change its own organization when new experiences demand it. A dynamic memory can learn." It is the way we deal with a new problem by observing new information to generalize new solutions from past experiences. By understanding a new problem, we can dynamically solve our new problem to reflect our experiences. A MOP is "information about how memory structures are ordinarily linked in frequently occurring combinations" (Schank 1982). It is an important approach in which past experience is structured. It values past experience which is not often integrated in the computing systems. These past experiences are used to interpret new inputs using the most closely related past cases.

Using cases in CBR, it is possible to provide better knowledge sharing and reuse solutions because CBR cycle involves revision and refinement phases. In general, there are two parts to a case (Aamodt 1994): the first part is the lesson it teaches, and the second part is the context in which it can teach its lesson. According to Aamodt (1994), "a case is a conceptualized piece of knowledge representing an experience that teaches a lesson fundamental to achieving the goals of the reasoned." Therefore, a case or problem situation can be defined as a conceptualized part of knowledge representing past experience.

2.5.1 Case-based Reasoning Architecture

In general, CBR refers to a problem-solving paradigm that relies on case representation, instead of only relying on general knowledge of a problem domain. Case representation in a CBR system includes a detailed problem description and a detailed solution description. Within a case representation, most types of data can be stored in a case. For example, stored data in a relational database, photographs, sound, and video can be represented in a case. However it may be difficult to represent large amount of inter-related data in a case. Therefore the functionality and acquisition of information need to be clarified first before deciding what should be represented in cases. Watson (1997) points out what information should be in a case using two pragmatic measures: the functionality of the information and the ease of acquisition of the information. In fact, CBR is dependent on the structure and collected case in case repository, so it is important to have a mechanism that organizes information that can be retrieved when it is required. Case representation

also should have a standardized mechanism that is supportable, suitable and appropriate to support case retrieval. There are four phases in the CBR cycle: retrieve, reuse, review and retain as shown in figure 2.5 (Aamodt 1994).

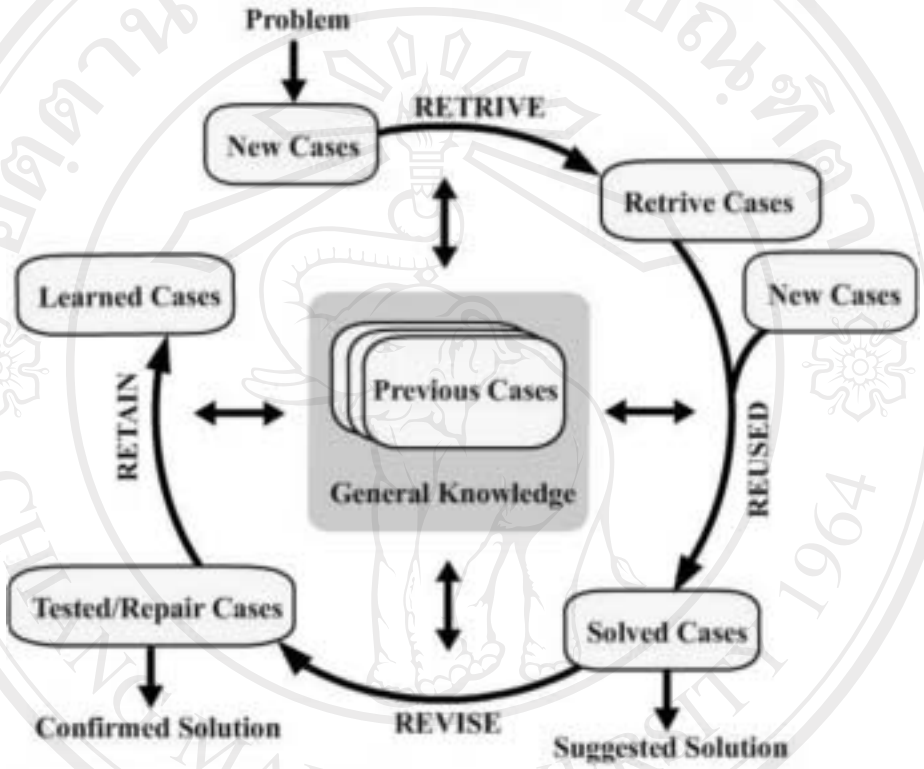

Figure 2.5 CBR cycle (Aamodt 1994 and Watson 1997).

The retrieval phase is to decide which case in the case repository is similar to a target case (target case is the current problem to be solved). When the case that is the most similar to the target case is found, then the CBR system retrieves the matched case that can provide a detailed solved problem description to the problem. There are techniques used to retrieve cases such as machine learning algorithm (Watson 1997), surface matching (Watson 1997) and a deeper understanding of the relationship

(Watson 1997), but the three widely used techniques of case retrieval are Nearest Neighbor Retrieval (NRR), inductive retrieval and feature measuring retrieval.

Nearest Neighbor Retrieval (NRR) is a technique used to measure how similar the target case is to a source case (Kang 2002). It processes retrieval of cases by using the comparison approach of a collection of weighted attributes in the target case to source cases in the CBR library. If there is no matched case in the CBR library, CBR system will return the nearest matched source case. The return of the nearest matched case can be found using the following equation (Kang 2002):

$$\text{Similarity}(T,S) = \sum_{i=1}^{n} f(Ti,Si) * Wi$$

where

T is the target case

S is the source case

n is the number of attributes in each case

i is an individual attribute from 1 to n

f is a similarity function for attribute i in cases T and S

W is the importance weighting of attribute i

The equation of the nearest neighbor retrieval technique represents the sum of similarity of the target case to the source case for all attributes multiplied by the importance weighting of individual attributes. The CBR system retrieves a

meaningful case that may provide a detailed solved problem description to a new problem. However, the nearest neighbor retrieval technique is not efficient. This is because whenever new cases are introduced, indexing needs to be performed and this could affect efficiency.

Inductive retrieval is a technique used to extract rules or construct decision trees from past cases (Aamodt and Plaza 1994). This technique processes a target case based on indexed source cases. The source cases are normally indexed by keywords and stored into a set of cases. The set of cases are divided into a decision tree structure. If the target case is not found in the decision tree at runtime, then the CBR system may not retrieve a source case. Aamodt and Plaza (1994) and Watson (1997) suggest the use of a combination of these two techniques in which inductive retrieval is used to retrieve a set of matching cases and then the nearest neighbor retrieval technique is used to rank the cases in the set according to their similarity to the target case.

Feature measuring retrieval (Watson 2001) is a technique used to compare features of the stored cases with the new case. A distance measure equation as shown below is used to measure distance between cases.

$$\text{WeightedDistance}(X,Q) = \sqrt{\sum_{f=1}^{n} (W_f \times \text{difference}(X_f, Q_f)^2)}$$

Where W_f is the weight assigned to feature *f*, and

$$\text{difference}(X_f, Q_f) = \begin{cases} |X_f - Q_f| & \text{if feature } f \text{ is numeric} \\ 1 & \text{if feature } f \text{ is symbolic and } X_f \neq Q_f \\ 0 & \text{otherwise} \end{cases}$$

After calculating the distance of each case, a ranking is done under the basis of the total distance score. The top case in a rank is expected to be the most similar case.

In the reuse phase, the solution from the retrieved case is used to solve the target case. In general, reusable case is more user-acceptable because its solution has already been accepted and convinced by the previous user. At the reuse phase, the solution from the matched case can be used without modification, or adaptation may be applied to adapt the solution to match the new problem.

Adaptation is a technique to alter the retrieved case to produce a new solution to a new problem. The solution of the retrieved case can be changed so that it can be presented to suit new use. The purpose of case adaptation is to improve the CBR system's overall problem solving ability using newly introduced cases for future use. The two most widely used techniques of case adaptation are: structural adaptation and derivational adaptation.

Structural adaptation is a technique to apply adaptation rules or formulas directly to the stored solution in the CBR library. Once a case has been applied by the

adaptation rules or formulas, the CBR system adapts the case as a match with the new problem. On the other hand, derivational adaptation is a technique used to reuse the rules or formulas that generated the original solution to produce a new solution to the current problem (Watson 1997). The retrieved solution then must be stored as an additional case in the CBR library so that it reproduces a new solution to the new case.

In the revise phase, the solution needs to be verified and evaluated to match the correctness of the solution. Once the verification is completed, the target case with its solution will be retained in the case memory. This is the retain phase of the CBR cycle. Indexing is commonly used in the case retainment phase in CBR. It allows retrieval of cases to be optimized. However, it is important that indexing be provided at an appropriate level of generality in terms of global and local context, so that it reflects the hierarchical structure of cases (Aamodt 1994).

There are a few considerations when building CBR systems. One of the considerations is the quality of CBR reasoning (Aamodt 1994). Generally, the more experienced the CBR system then it is more likely to perform better than the one that is less experienced. In other words, the ability to understand a new problem in terms of old experience has important role in the CBR systems. This is because more experience CBR system has more successful ability to propose solutions that are stored in the case repositories to solve new problems. On the other hand, less experience CBR system may not be able to propose solutions to new problems during the reasoning processes. In addition, the adaptation ability in the CBR systems is

important, especially when it attempts to use an old solution to fit in to the new situation. The ability to integrate new experiences into its case memory appropriately is important so that solved cases are achieved for later reasoning purpose. Evaluation or repair needs to be achieved to provide feedback to other similar cases. The CBR system must learn from its experiences in such a way that it integrates mistakes and it is able to tell us what was wrong or right when new problems repeated the next time.

2.5.2 Examples of Case-based Reasoning in Real World

Various CBR systems have been built over the last twenty-five years. As discussed, CBR starts with Roger Schank's dynamic memory and MOPs. The Computerized Yale Retrieval and Update System (CYRUS) is the first CBR system that is built based on Schank's dynamic memory model and MOP theory (Schank 1982). The system allows users to ask questions about the travels and meetings of former US Secretary of State Cyrus Vance (Watson 1997).

PROTOS is a case-based classification system and uses knowledge acquisition approach to handle a problem of classifying auditory diseases (Kolodner 1983). Domain knowledge of hearing disorders is represented as cases. PROTOS uses a trial-and-error method of problem solving. It is known as a heuristic approach that learns, discovers and solves problems on its own by trial and error (Kolodner 1983). When PROTOS faces a problem, it tries to find a solution. If a problem is successfully solved in the first trial, PROTOS does not learn domain knowledge of hearing disorders that is represented in the new case. However, if PROTOS solves

the problem in the second or third trials, PROTOS learns the cause of the particular case during classification. If a user requires further modification and explanation, it can be stored in the new case for future usage.

Another example is CASEY (Kolodner 1983) which uses case-based and model-driven approach to complement the CBR process. It is a case-based and model-based reasoning approach because "when a problem turns out to be unsolvable by retrieving a past case, a general domain knowledge model is used in a second attempt to solve the problem". CASEY is used to analyze patient symptoms. For example, where two patients show differences in symptoms, then it has to be reconciled by CASEY's evidence rules. The rules are applied and used as part of the model. The evidence rules examine previous diagnosis until it matches. Especially, rules go on to create an explanation of one patient's symptoms by adapting the other patient's diagnosis.

HYPO is another example of CBR application which is used to produce and assess arguments for both the defendant's and the plaintiff's side in the domain of law (Kolodner 1983). It is case-based argumentation where it compares and contrasts procedures in reasoning. If several different cases are available to make legal argument, it provides an idea of which case is preferable in terms of taking the strongest arguments. The contribution made by HYPO is that it shows some of the steps involved in the cognitive processes and the knowledge in engaging such reasoning.

2.5.3 Examples of Case-based Reasoning in KMS

In literature, CBR has been applied successfully in various KMS to reuse previous solutions to resolve new problems. The application of CBR techniques allows KMS to acquire new knowledge, by adopting knowledge gained in the new cases and reusing the old ones. This allows new knowledge to be shared and added in the knowledge repository. In this section, we provide some examples in which CBR has been successfully applied to KMS. The first example is British Airways. British Airways needs a diagnostic tool to support maintenance technicians in solving problems in Concorde Olympus power plant. Due to the complex assembly procedures and the need of cost effective operation, British Airways requires a repository of diagnostic experiences that can be made available to all technical engineers (Magaldi 1999). A software package called SportLight from CaseBank Technologies Inc is used to develop the KMS. It is the CBR software to support the troubleshooting of complex equipment, systems or processes (Casebank Technologies 2002). One of the intangible values identified is the knowledgeable assets gained by the organization.

The second example is the World Bank. The World Bank needs a knowledge-centered mechanism to utilize economic and social development projects. It aims to provide cooperated knowledge repository to lead a better search and browsing mechanism in order to make decisions and judgment that exists in the collection of related experiences (Moussavi 1999). Problems, relevant useful information and solutions are stored in case library. Then it is indexed between similar cases. A

situation assessment user interface is developed to help the end user to search the cases. In this example, CBR improves the quality of operational values in the World Bank.

The last example is The Great Lakes Geriatric Interdisciplinary Team Training (GITT) project which is a collaborative research to support the long term care of Alzheimer's disease patients (GITT 2002). In this example, CBR is not only used to solve problems for the patients' symptoms but it is applied to support GITT participants in decision making when conflicting perspectives arise.

2.5.4 Summary

In summary, CBR refers to a problem solving paradigm that relies on case representation, instead of only relying on general knowledge of problem domain. When the new problem issue arises, the retrieval process identifies the problem as a case to find out the most similar one in the past cases. Then, if there is any matched one in the past cases, it will be presented as a solution of new case. If it is necessary, adaptation occurs and a new case is created.

The CBR model has some advantage over the other AI approaches and the CBR approach is also used to build intelligent systems which increasingly domain of knowledge. Then the CBR can help librarians to solve the problem of various domain metadata cataloging by reusing the previous solution of metadata cataloging with the new problem.

2.6 Knowledge Management Systems

We begin this section by first defining terms that include data, information and knowledge. Data are the raw inputs of individual facts, statistics, or items of information. Information is processed or value-added data and knowledge is the understanding of information meaning. Knowledge can be classified as tacit and explicit knowledge. Tacit knowledge is knowledge that cannot be easily described such as skills, experience or native talent. Explicit knowledge is skills and facts that can be written down and taught to others such as technical documents.

Since 1970s, knowledge started to play an important role in organizational strategy. By the 1980s, the importance of organizational knowledge is increasingly recognized. Organizations have focused on processes and strategies to manage innovation and to build knowledge. As a result, a system is developed to provide a technological base for managing knowledge. The term knowledge management can be seen as management of knowledge related activities. These activities include broad, multi-dimensional and covers most aspect of the enterprise's activities (Wiig 1997).

There are different approaches in which knowledge management is used in the organizations. The first approach is that of the repository model. It focuses on managing information and reusing knowledge in concrete formats (Turban and Aronson 2001). Knowledge management is also to be "making a direct connection between an organization's intellectual assets". This approach is viewed as a legal

approach. It involves intellectual capital, copyright, patents and trademarks (Turban and Aronson 2001). Knowledge management can also be regarded as business intelligence. It is a process to produce valuable up to date information for operative and strategic decision making (Turban and Aronson 2001). Other approaches include the cognitive and continuous learning approach which is the ability to learn (Turban and Aronson 2001). It involves an individual ability's to acquire continuous and ongoing renewal of organizational information and reuse it for problem solving and decision making. In addition, cognitive approach of knowledge management focuses on learning within groups as well as individual's learning level. Then it can be seen that a lot of organizations have begun to recognize knowledge as the most valuable assets in their organizations. These valuable assets include personal skills and experience as well as any stored information in the organizations. In general, KMS refers to a system designed specifically to provide the sharing and integration of knowledge (Walsham 2001). It allows corporate knowledge to be shared in the organizations effectively and efficiently (Walsham 2001). According to Watson (2001), KMS is a system that can provide competitive advantage by giving decision makers the necessary insight into patterns and trends affect their domain. A KMS is able to make comparisons, trends analysis, and historical and current knowledge presentations. But the most important thing is KMS enables decision makers to analyze and understand the patterns quickly and identify the most significant trends. From this perspective, KMS provide essential knowledge and related information to decision makers in making better decisions.

To increase the effectiveness of the knowledge development process, it has been suggested that “organizing strategies should be defined and initiated based on knowledge development phases” (Ganesh 2000). There are four phases of knowledge development cycle, which includes knowledge creation, knowledge adaptation, knowledge distribution and knowledge review (Ganesh 2000). The knowledge creation phase involves proving, learning and evaluations of common means of managing the knowledge, and the knowledge adoption step involves the use and re-use of existing knowledge. Then the knowledge distribution phase provides ease of access, sharing and manipulation of knowledge via knowledge infrastructure, media selection and knowledge-fundamental. Finally the knowledge review and revision steps involve testing processes of validity and reliability.

A similar process is found in the KMS development life cycle. A typical KMS development cycle consists of create knowledge, capture knowledge, refine knowledge, store knowledge, manage knowledge and disseminate knowledge (Turban and Aronson 2001) (see figure 2.6).

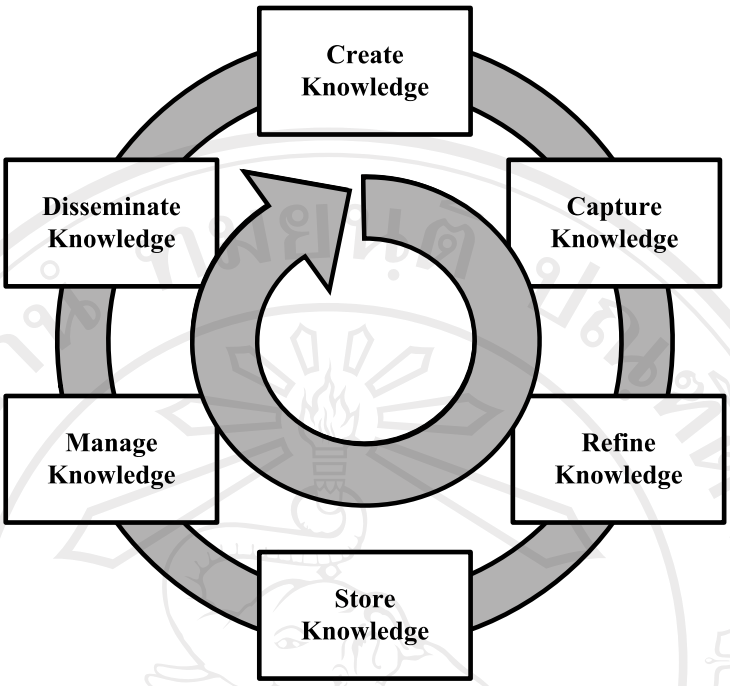

Figure 2.6 KMS development cycle (Turban and Aronson 2001).

Create Knowledge is the process of facilitating the solution of a problem in different contexts. Capture Knowledge is the process of driving and adding value to an organization from generated data and information. Refine Knowledge is the process of placing knowledge in specifying tacit to explicit knowledge. Store Knowledge is the process of accumulating knowledge in a repository. Manage Knowledge is the process of reviewing and revising knowledge. Disseminate Knowledge is the process of making knowledge available, distribute and share throughout the organization.

In particular, the knowledge creation phase is the most important phase and is often referred as knowledge acquisition. Knowledge acquisition is a process of translating implicit knowledge into explicit form. According to Partridge and Hussain

(1995), “knowledge acquisition is a very labor-intensive activity. It is almost an art-form with questions arising for which there are no algorithms or computer programs”. Extensive researches have been conducted on ways to improve the knowledge acquisition process. In particular, attempts have been made to either partially or fully automate the knowledge acquisition process using Artificial Intelligence (AI) techniques (Partridge and Hussain 1995). Partial or semi-automatic approach allows the interaction between knowledge experts and knowledge engineers to be reduced. On the other hand, the automatic approach refers to using AI technique that allows the experts to build their own knowledge bases without the assistance from the knowledge engineers.

The popular uses of the World Wide Web (WWW) and Internet technologies help to speed up the process of knowledge acquisition. Knowledge engineers often have to capture knowledge from experts using interview method, which is often time-consuming. To speed up the traditional manual method of interviewing, a knowledge engineer can interview experts via electronic interviewing. Documented knowledge can be submitted via electronic forms and these forms can be retrieved and stored in the knowledge base. In practice, video-conferencing and web meeting technology can be used to support people in the knowledge communities to share tacit knowledge. Other manual knowledge acquisition methods include tracking and observation. The tracking approach is used to find what information is being used and how it is being used. However, these manual knowledge acquisition methods are slow and prone to error.

According to Partridge and Hussain (1995), the knowledge acquisition process starts with planning knowledge base or knowledge repository in KMS. It organizes knowledge for the knowledge base, followed by knowledge extraction from the different relevant sources of knowledge. Then, it formulates and represents knowledge for inference making. For example, a decision table and production rules are used to express logical relationships, and to identify set of conditions and actions. After encoding knowledge in machine-readable form, implementation of knowledge base is followed. When the knowledge base is ready for testing, knowledge engineer and the domain expert will verify and validate it to ensure the systems have met the requirements. Finally, it is ready for systems test.

In summary, the purpose of building a KMS is to share corporate knowledge in the organizations. Moreover KMS is used to support for knowledge sharing, within Communities-Of-Practice (COP) is a valuable focus for contemporary organizations. One of the ways of developing successful KMS is through facilitating the concepts of knowledge sharing and use in practice. However, the traditional development of KMS is considered to be weak in facilitating the concepts of knowledge sharing and reuse in practice. Making reuse of previous useful solutions to solve new problems by referencing to old solutions is often not easy to achieve. In addition, gathering well-agreed terms of reference in the communities of practice, it may not be easy to use knowledge in organization.

Therefore, KMS can be considered as a linkage tool between Thai domain experts and Thai librarians via the internet. The purpose of building a KMS is to

share knowledge in the organizations then Thai domain experts and Thai librarians can be used KMS as a tools for sharing knowledge by revise the proposed solution together.

2.7 Conclusion

To ensure knowledge sharing and reuse can be achieved in KMS, especially in a networked environment such as the WWW, CBR techniques can be applied to provide an opportunity to allow new knowledge to be updated, stored and retrieved in the KMS. CBR techniques can be applied to knowledge management to provide effective problem solving solutions, creating rich knowledge repositories, and decision supporting mechanisms. It can be used to support knowledge management tasks.

Knowledge management can be viewed as a process of multiple activities. A design of CBR system can be embedded to retrieve and adapt knowledge management activities. CBR can be used to improve knowledge acquisition processes of KMS by allowing new knowledge to be updated and learned. To support CBR, a need of a well-defined set of domain interest in community has to be presented clearly to prevent ambiguity. Metadata provides an opportunity and a feasible approach used to conceptualize a set of terms in the community of practice. Metadata provides new opportunities to prevent ambiguities in knowledge representation by supporting well-agreed terms or vocabularies. This gives better support for knowledge acquisition

processes. To support a standardized platform for creating human understandable format, the information extraction features can be used.