
↓

Cognitive Based Context Aware Reference History Management Tool

↓

Dharani Punithan*, Bob McKay**

Abstract The aim of the research is to focus on the cognitive principles and to achieve human-level intelligence in referring context based browser history and the Windows history. One of the major problems faced by today's computer users is insufficient and single exclusive context based reference of the browser history and the Windows history. Today we search for the browser history and Windows history in different places even though the context is the same. For e.g., When working on a research paper or preparing a business presentation, a user may require to refer many web sites on the internet and various documents on the local computer. The browser can provide only time based history. The windows document history is also time based and limited to list only few documents. Hence, we propose a tool "*Cognitive Based Context Aware Reference History Management Tool*" which helps to access the exclusive reference of context and time based history in one place. The tool also proposes to store image history with urls and classifies images of a specific topic accessed in different time, bookmarks management and cross browser history management. These features are very useful as we can access all related documents (doc, docx, ppt, pptx, pdf, txt, and html), web pages, images and bookmarks in one place. The tool uses the cognitive principles like classification and association to achieve the purpose.

↓

Keywords: *Cognitive Science, Human-level intelligence, Context Awareness, Win32, Win32 API*

This work was supported by the Brain Korea 21 Project and the Korea Research Foundation Grant funded by the Korean Government (2008-KRF-313-D00943).

*주저자 : Seoul National University, Computer Science Engineering, PhD Student e-mail: dharaani_punithan@msn.com

**교신저자 : Seoul National University, Computer Science Engineering, Associate Professor e-mail: rimsnucse@gmail.com

1. Introduction

Capturing and storing the history of user activities is an important requirement for computer systems. In particular, history information on previously accessed documents and browsed web pages are useful in avoiding the frustrations in searching or browsing the Windows History or Browser History. However in the existing systems, the recent history management is not efficient. Most of the time, the user tends to search and browse the history again and again. Moreover, the traditional way of opening the documents are tedious if the folder hierarchy is deeper. Sometimes, we need to traverse a long tree structure to open the required document. If we want to access files on same context and if they are cluttered, it is still more painful to access all the documents to have a single exclusive view.

With existing utilities such as *browser history* or *recent documents* on windows systems, when users work on large number of documents or browse large number of web pages on a various topics or subjects over the period of several weeks or months, it is not possible to check the history information based on a specific topic or subject of interest. This is because of the reason that those utilities provide only time based history information. Moreover, such utilities are type specific and system dependent. For example, history information utility (recent documents) in MS Word lists the history of word documents only and not other types of same topic or subject. These utilities are OS dependent and do not carry the history information across the operating systems. There is also no cross browser history management.

Though there are tools like X2Net Recent Documents [1] available in the market to access the recent history, they are only based on date or application type. They are not context based on the topics of interest.

In this paper we propose a new concept and overall system architecture to build a cognitive based context aware reference history management tool which helps the user enormously to get a single exclusive view on their activity history. We also show the proof of our concept by developing a sample tool with minimum features on windows system for simplicity. However designing generic/system independent tool is implementation specific and it is left to the developers. We also conclude that the various usages of our architecture and scope of work that we have planned for future.

2. Overall System Architecture

The below diagram depicts the overall system architecture for Cognitive Based Context Aware Reference History Management Software. Here we define the functionalities of various modules required for building a Context Aware History Management Software in general.

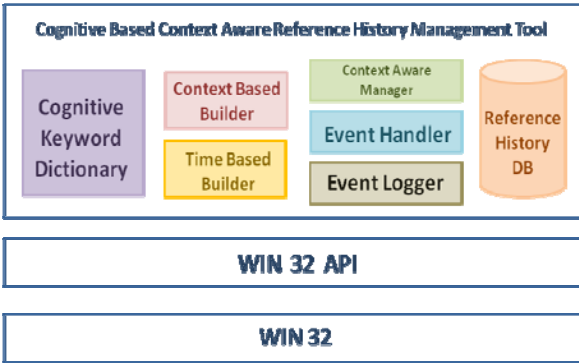


Fig. 1 System Architecture

2.1 Cognitive Keyword Dictionary

This module is the foremost and important part of the system. It maintains large number of keyword groups on various topics of interest. Keyword dictionary is built by either cognitive learning methods or semi-cognitive learning methods or both. Keyword dictionary construction is the responsibility of Context Aware Manager.

In Cognitive learning methods, Keyword Dictionary construction is dynamic in nature so that more and more keywords can be added or existing keywords are regrouped to the dictionary as the user access or browse more and more documents or web pages.

In semi-cognitive learning, user initially creates a topic list of interest through user interfaces. Context Aware manager in turn should search and collect as much as keywords from various internet resources such as Wikipedia, Thesaurus etc.

Example construction of Keyword dictionary by cognitive learning method is shown below.[2,3]

Learning cycle 0:

Main Topic Name: **Machine Learning**

Learning cycle 1:

Main Topic Name: **Machine Learning**

Sub topic: Artificial Neural Networks: AI, expert system
Sub topic: Genetic Algorithm: Evolutionary Computing

Learning cycle 2:

Main Topic Name: **Machine Learning**

Sub topic: Artificial Neural Networks: AI, expert system, intelligent retrieval, knowledge engineering
 Sub topic: Genetic Algorithm: Evolutionary Computing, fitness function

Learning cycle N:

Main Topic Name: ***Machine Learning***

Sub topic: Artificial Neural Networks: AI, expert system, intelligent retrieval, knowledge engineering, machine learning, natural language processing, neural networks, robotics
 Sub topic: Genetic Algorithm: Evolutionary Computing, fitness function, crossover, mutation

2.2 Context Aware Manager

The main functionality of Context Aware Manager is to cognitively construct keyword dictionary and update the keywords in the dictionary from user activities of browsing and accessing of documents. It is also responsible for classification of accessed/browsed information based on topic and time.

Context Aware Manager is called by *Event Handler* module when a user accesses a website or opens a document. It searches the currently opened document/web pages and extracts key words from them. The extracted keywords are added/updated in the Keyword Dictionary cognitively based on the assigned weights to them. This module also builds and maintains Classification mechanism based on context (topic) and time. This Classification mechanism is used to group the documents/web pages from event logger and stores in Reference History Data base, based on the topic or time.

The strong association of the keywords plays an important role in classifying the history information of the web pages and the documents. The association between the keywords is decided by the weights assigned to them during the learning process. The weights are determined by the frequency of the keywords. Also the association helps to avoid unnecessary pages and documents which are accessed during the main task. Because the user may not access only the web pages or documents related to his work. The user's mind may deviate from his work for relaxation. Hence Context Aware manager should intelligently avoid those pages' or the documents' history.

Classification mechanism is implementation specific. Hence developer is free to design the classification mechanism as per his requirement.

2.3 Event Handler/Event logger

Event Handler monitors the user activities such as browsing and document access continuously. Whenever a user opens a document or searches information using

the browser, it captures the events and invokes Event logger. In turn, event logger stores the event information in its local database with information such as name, type and time. Context aware manager will use this data base for Context (topic)/Time based listing. Event logger module is also implementation specific. The event logs are comma separated values and are de-serialized using jsefa [4] APIs.

2.4 Reference History Data base

The Reference History Database is constructed using file systems. It consists of various directory structures/folders created by Context Aware Manager based on topics. These directory structures/folders purely based on topics and subtopics. The actual context based contents are also comma separated values and are de-serialized using jsefa [4] APIs.

3. Tool Description

Based on the system architecture described above, we have implemented a Trivial Context Aware Reference History Management Tool as proof of concept. The tool is developed on Java platform. This is implemented based on semi-cognitive learning for simplicity.

In this approach, we need a *Terminology list* which consists of the initial information of few topics (3 or 4) of user's interest. At first, user enters few topics of interest (semi-cognitive method) as base data in the Terminology list through the user interface provided.

During Initialization, Context Aware Manager module passes the Terminology list information to the web crawler which in turn fetches all the related keywords of Terminology list from Wikipedia and builds the keyword dictionary. The dictionary is based on key-value pairs. The Wikipedia page topic is the key in the dictionary and the links in the Wikipedia page are the values in the dictionary. However, in cognitive learning methods, the keyword dictionary should be built from user activities of browsing and document accessing over the period of time. For simplicity, we have implemented only the semi-cognitive learning method. But the keyword dictionary is updated by adding new terms. This is done when the document or web pages are searched and matched with the any of the keys or values in the dictionary. The updation of the dictionary is allowed if the frequency of a particular term is high and the term is not present in the dictionary.

All the user activities of browsing the web pages and accessing the document are captured and registered in *Event logger*. Event logger may capture the time, name, and hyperlinks (file path in case of a document) of various documents or web sites accessed. In our implementation Event logger information are collected from *index.dat*[5] system file using Java classes. We

also have checkpoint mechanism for not processing the same logs again and again Other Windows low level components are accessed using SWT Win32 Extension [6]. Below depicts the software architecture of the tool.

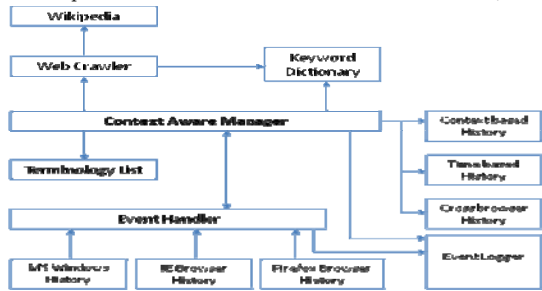


Fig. 2 Software Architecture

Context Aware Manager module then checks each entry (document or web page) in the Event logger list and searches for the existence of keywords of each topic from keyword dictionary. The text extraction from the documents (doc, docx, ppt, pptx) is done with Apache POI API [7]. The text extraction of text file is done with FileUtils of Apache commons API. The text extraction of pdf is done with PDFBox [8]. The html text extraction is done with java swing components. If there is a match between topic key or keyword list of a topic in the dictionary and extracted document or web page text from Event logger entry, for each such matching, a weight is added. As number of matching increases the weight also increases. Based on the value of the weight, Context Aware Manager creates a folder with matched topic name and adds the corresponding entry into the folder.

Context Aware Manager stores all the history information based on the context and the time in the Reference History DB. The information in the reference history database is later accessed by the context aware manager when the user needs to know about the reference history information.

The below is a sample UI of the tool which lists all the web pages accessed by the user on Artificial Intelligence. The UI is developed using the Standard Widget Toolkit (SWT) [9].

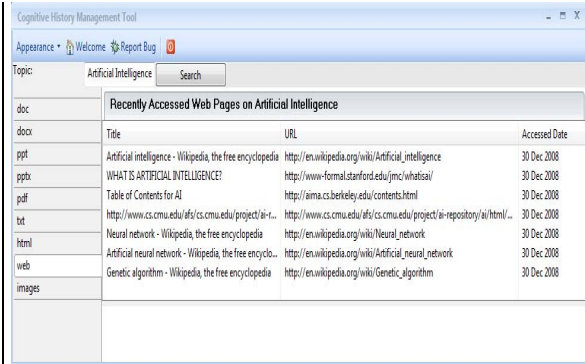


Fig. 3 Tool's User Interface

4. Conclusion

Cognitive based Context Aware Reference History Management Tool overcomes the problems faced by the users when they want to access the exclusive context based browser history and the Windows history. Using the proposed architecture the user can access an exclusive reference history of his activities on a particular task which means all history information in one place. Thus it helps to reduce the frustrations in searching for the information in browser history and in the Windows history.

5. Future Scope of Work

The existing Cognitive based Context Aware Reference History Management Tool is developed based on semi-cognitive methods just for simplicity. In future, we develop fully automated cognitive tool. We also plan to extend the Context Aware History Management system for context based image history management, book mark management and cross browser management. The current system works only on Windows Vista. Hence we also plan to define porting layer in future for platform independence.

6. References

- [1] X2Net Recent Documents, <http://www.softsia.com/X2Net-Recent-Documents-download-6vn5.htm>
- [2] B.-T. Zhang, "Cognitive learning and the multimodal memory game: Toward human-level machine learning", IEEE World Congress on Computational Intelligence, 2008.
- [3] B.-T. Zhang and C.-H. Park, "Self-assembling hypernetworks for cognitive learning of linguistic memory", Proceedings of World Academy of Science, Engineering and Technology (WASET), pp.134-138, 2008.

- [4] JSefa (Java Simple exchange format api),
<http://jsefa.sourceforge.net/>
- [5] Index Dat Spy,
http://www.stevengould.org/index.php?option=com_content&task=view&id=47&Itemid=88
- [6] SWT Win32 Extension, <http://www.swtui.cn/>
- [7] Apache POI, <http://poi.apache.org/>
- [8] PDF Box, <http://www.pdfbox.org/>
- [9] SWT: The Standard Widget Toolkit,
<http://www.eclipse.org/swt/>