Information Collection in Creative Design Process

MUHAMMAD IKHWAN JAMBAK[1] AND INDAH PERMATASARI[2]

[1,2] *Universitas Indo Global Mandiri*
[1] *jambak@uigm.ac.id*
[2] *Indah@uigm.ac.id*

Compiled March 10, 2023

Information is involved in the design process, and in fact, designing is an activity of converting data, information, and knowledge into a map and guidance of creating an artifact. Unfortunately, the existing instruments supporting designers39; information search are not fit for them in terms of their natural way of doing so. This research aims to develop an information search application purposely for designers, especially in the creative design process. And this paper will focus on developing and implementing a so-called Search Result Side Note application.

http://dx.doi.org/10.1364/ao.XX.XXXXXX

1. INTRODUCTION

Large amounts of data, information, and knowledge are used during a design process, especially during the earlier stages of design. Although they are abundantly surround the designers, however designers often neglect these resources and rely on discussions with colleagues in its place. At least there are three reasons to explain this phenomenon. First of all, the past research proves that the difficulties of utilizing these resources mainly due to the large amount that difficult to be handled by human beings and the varying in size, the differences in type as well as coming from different domains that too difficult to be structured in a computer based tool [**?**]. Secondly, it is due to the absence of quot;contextquot;, namely how to bring the right data, information, and knowledge into a design context for a particular design process. Thirdly, it is attributable to the incompatibility of the current search engine with the way designers searching and handling data or information. In fact, most of these internet search engines are typically intended to be used once during an information search process, while designers search for design information iteratively in a divergent (broadly with minor detail in order to get idea or creating choices) or convergent (very deeply and narrowly with detail in order to make a choice) search process. There are two more reasons why these search engines are not really fit to the design process, especially in the early phases. To begin with, these search engines assume the information need that is reflected in every search term (or keyword) of an information query, is mature, while in the early phases of design, the concepts are vague and the problems are ill-defined. Therefore, the information needs themselves are subject to be found. Lastly, because queries are not connected, the use of the available search engines at this moment does not give designers the opportunity to treat and utilize the contexts that have been built during the search process. Instead, those contexts are implicit and stored mentally. As part of of our research to solve the mentioned problems above regarding designers information search, an application has been developed and implemented. This paper is aimed at explaining an application that works over an Internet search engine in order to fit it into designers way of looking for and utilizing information for their design works. In section **??** , this paper introduce the basic theory of design and its related terminologies. Next, in section 2 the way how a designer in a creative design work looking for and utilizing information is briefly explained. Lastly, in section 3 the application that called as quot;Search Results Side Notequot; and it39;s development and implementation has been explained.

2. INFORMATION SEARCH IN CREATIVE DESIGN PROCESS

One of the challenges in supporting designers information search in a creative design process is that the amount of data, knowledge and information to be handled by designers is often too large and too broad as well as different in type and size that make it unmanageable to be implemented into a well-structured system. Nowadays, the enormous growth of web sites has opened up opportunities for designers to retrieve a broad variety of design information from the Internet. The existing search engines do offer not only better algorithms, but also technology advancements that potentially help designers to gain necessary information. Merely to help designers, these search engines need a keyword or phrase in order to retrieve the information. In other words, you need to know what are you looking for. In contras, in a creative design process that characterized by ill-defined problems, vague solution concepts and the ambiguity of possible keywords, designers might be in a situation where getting access to the correct and necessary knowledge or information for a particular design process is not always an easy task while data, information and knowledge are abundantly available and surround them. In the previous research, it was observable that designers do the information search in convergence and divergence ways concurrently. Designers generally use divergent approaches which involve the suspension of judgments that meant for collecting alternative choices or solutions. Meanwhile, designers use convergent approaches to evaluate or select those alternative choices or solutions. For the purpose

of this research, the author propose the designers information searching model as can be seen in Figure **??**. This model then become the foundation of the developed and implemented application that will be explained in section 3.

3. SYSTEM DEVELOPMENT

This section is divided into three subsections, they are user specification, system and interface specification, and the implemented Search Result Side Note application.

A. User Specification

The users of this system are designers who have minimum experience using internet search engines. It is assumed that the users need to complete one or more iterative information searches on the internet, and need to manage the results and the information search processes in a constructive way. It is also assumed that the users will complete individual or collaborative design projects. The users might be willing to use the results and the processes of the informational search in other projects or share this information with other users.

B. System and Interface Specification

From the system39;s conceptualization, literature study, and the results of the context knowledge study in conceptual design, the specific implemented system is as follows:

1. The implemented system is built on existing web search technology and operates in conjunction with a common internet search engine. The idea of this work is not to develop a new search engine. Instead of replacing existing advanced search engines, this system will amplify them to fit the nature of the designers, as well as their projects and purposes. As a prototype, the implemented system will utilize a browser technology component and a popular search engine that supports the chosen programming language.

2. The implemented system will support the user in building representations of one or many contexts, where each context is a structured set of keywords that contains Uniform Resource Locators (URLs), pictures, and supportive files, such as office application documents and media files. With the phenomena of design in practice, design problems needing to be solved or design ideas needing to be elaborated may occur in an iterative information search process using more than one search term. Each of these keywords may show one or more URLs as a result of the particular search. Each of the search terms may also be supported by the user39;s existing local computer files, such as office documents, pictures, and media files.

3. This system will provide an overview of all available contexts in a particular design project and also provide the graphical presentation of any contexts. One way to make the context explicit is to represent it in a graphical representation.

4. This system enables the editing of any context; keywords can be included or deleted, and local or newly retrieved documents can be associated with keywords.

5. This system encourages the designer to search for information from a context selected from existing sources rather than from keywords or stored bookmarks.

6. This system encourages the designer to spend more time and effort on enriching existing contexts and/or creating new ones.

7. This system enables the copying of contexts from preceding and/or shared projects. It also enables the copying of entire projects (as a starting point of a similar design or a redesign).

8. This system enables the designer to make his/her project sharable with other designers.

From the system requirements specified above, a list of interface requirements is drawn as following:

1. The search application interface should contain a common search engine.

2. The search application interface should allow users to see the list of all queries that have been given by the search engine and show the results.

3. This system should allow designers to decide for themselves the relevance of each result.

4. This system should allow the designers to decide for themselves which keywords are within a context. The user should be able to edit the context itself in terms of adding or removing relevant keywords.

5. This system should allow designers to find and attach their local supportive files onto a particular context.

C. System Implementation

An application has been implemented using a *Visual Basic*and *Java* programming language. The system offers the possibility of managing the queries and results in a structured way. and its capabilities to coexist with an existing search engine.

C.1. System Database

The *MSAccess* database management system has been chosen to implement this information design. The numbers on the tables that have been created are as much an entity in information design as any other component, but some details to the attributes have been added. The users and project tables play a key role in determining how to personalize the search process. The combination of the primary keys from these two tables is to ensure that each information search process carried out on the internet is unique. Since the relationship between these two tables is many-to-many, where a user can have many projects and a project can belong to many users (for example, when they are developing the design project together), a table called quot;worksquot;, is a kind of index allowing the query to be unique and expectantly faster.

C.2. Search Module

This module is at the core of the implemented system. In this module, an existing search engine is utilized. The user will use the search engine as usual. Any of the features available from the search engine will be available to the user. In addition, this module will capture the query text (keywords, combination of keywords, or sentences) and write it into a “history39;39; memory

before it is passed to the search engine running on the Internet browser. When necessary, the text in the memory can be passed to the search engine. In return, the search engine will display a list of URLs as a result of the passed query. The memory will

be flushed whenever the module is closed. Instead of going to the location of the clicked URL, a pop-up window will appear to show the Internet site. It is then the decision of the user to determine the relevance of the site. If the user decides to keep the chosen Internet site, both the query text and the URL text will be stored in the "Keywords39;39; and "Bookmarks39;39; tables. Whenever a query text is loaded from the database, it will be passed on to the search engine. In addition, all lists of internet sites, bookmarked URLs, pictures, and documents will be loaded.

C.3. Search Engine Utilization

To utilize a search engine, a web browser component and text box must be placed in a *visual basic* form. Whenever the search module is executed, this form will be loaded and will call the web browser component. The empty web browser component will be initiated by a chosen search engine web address and then locked to avoid a direct search through the search engine. The search term typed by the user in the textbox will be taken as a search argument value. After encoding this search argument into a form that is accepted by the chosen search engine, a complete URL that contains the search engine address and search argument will be composed. This URL will be called by the activated web browser by using the quot;Navigate2quot; method. In return, a list of search results will be shown by the web browser. The general algorithm of this process and the example of the search engine utilization subroutine can be seen below. Unlike using the search engine directly, this system will not redirect the web browser to the clicked search result, but rather will load a pop-up window called the result browser window that consists of a web browser. The user can still see all of the listed search results while checking a clicked search result. After checking the search results in the result browser window, the user can decide to bookmark the result or not. More details about bookmarking activities will be described in the next subsection. The implemented system should represent the search engine as it is and should allow the user to see and use all available features that belong to the search engine and web browser. In other words, the embedded search engine39;s features must be available for use. In order to differentiate between the click on a search result that triggers the loading of the result browser window and other clicks that enable the use of a search engine39;s features, an quot;intelligentquot; check procedure has been added. Only a double click on a URL listed in the search results will trigger a result browser window to appear;otherwise, it will be considered an action of the search engine or web browser.

C.4. History and Bookmark

The search terms written in the text box are not only passed to the search engine, but also have been sent to a temporary history list as shown in figure **??**. This temporary memory is used to catch user39;s "trial-and-error39;39; and "mind jump39;39; from one topic to another, as a tool for the user to keep track of his/her searches. Even if these search terms are right, relevant,

"wrong39;39; or "irrelevant39;39;they could always be revisited and be selected again in which system will pass the information to the search engine shows results again. Technically, each time the search module sends the search term text from the text box to the search engine, it sends this text to a listbox as well. A double-click on one of these listed search terms will activate the search engine to search for information on the internet and present the search results based on the clicked text. Since these search terms listed in the listbox are represented as "short memory39;39; in this information search process, they will be cleared whenever the search module is closed. Contrastingly, the bookmark system is a representation of a "long term39;39; memory. When the user bookmarks one of the search results, it means he/she affirms that the bookmarked search result is relevant to the design process, and therefore both the search term and selected search result need to be stored in the system database as an intellectual asset. Before showing the selected URL, the system will first check if the search term has been previously stored. The system will only store the search term once, when the first URL from a particular query is bookmarked. If the search term does exist in the database, the system will then check to see if the URL has been stored before as a particular associated search term. If so, then the system will only show the result and will not complete a pre-store process. Otherwise, only a pre-store process for a URL, or for both the search term and URL, will be executed when the user clicks the quot;bookmarkquot; button. In the pre-store process, the system will access the "keywords39;39; and "bookmark39;39; tables of the CDIRS database and collect sets of records called recordsets from those two tables by executing Sequel Query Language (SQL) commands. The system will then do a quick seek to find the search term. If the search term does not exist, then the system will create a unique identification (ID) for the search term and for the URL, or the system will only create an ID for the non-existing URL.

The system will then start a data store process to write the search term ID, search term text, URL ID, and URL text to the recordset. By using the data transaction methods, the system will only physically transfer the data to the database if there is no error and the process can be completed; otherwise it will cancel the transaction. After a successful transaction, the recordset will be closed and the memory will be released. By following this procedure, the system can keep the data integrity in the physical database. When a bookmark has been successfully created, the search term will be displayed in a combo box (a list box that only displays a single item at the first item index). Only the latest stored search term will be displayed in the combo box, while all relevant URLs will be displayed in a list box as shown in figure **??**. Every time that a URL from the list is clicked, the information (web site) will be displayed in a pop-up window screen. Each one of the stored search terms in the combo box is linked with the search results, the list of bookmarked URLs and the list of a user39;s relevant documents, such as electronic pictures/drawings, MS Office files, and multimedia files. Once the user selects the combo box and chooses another search term,

the search engine will update the search results based on the chosen search term and a list of bookmarked relevant URLs from the particular displayed search term will be shown. The user39;s embedded files will then be listed on a multi-tab pad as can be seen in figure **??**. Essentially, every stored search term builds a very low context definition, namely, a connection between a search term, relevant search results, and a user39;s relevant files from the "outside39;39; of the system environment. More details concerning a user39;s documents will be given in the next sub subsection.

C.5. User39;s Embedded Documents

One of the fitures of this implemented application is that it allows the user to link electronic documents to a particular search term. The assumption is that the user might want to use the same keyword to find information on the internet or to find his local document. The other assumption made for this system implementation is that the user might want to enrich information

from internet search results with his local documents or vice versa. These embedded documents might be available in many formats, such as a note, a drawing, a physical model, a book etc. However, it is possible to transfer this format to an electronic format. This can be categorized into three types, namely, pictures, office documents, and multimedia (sounds and moving pictures) as shown in figure **??**. In order to embed the documents into the system, a browsing menu has been created. This menu will appear as a pop-up window when the "add picture39;39;, "add document39;39; or "add media39;39; buttons in the multitab pad are clicked. Basically, this browsing menu contains browsing tools like "Explorer39;39; in MS windows, where the user can define the "drive39;39; and "directory39;39; of the user39;s document file as illustrated in figure **??**. To quickly view a selected file, the user must click one of the files listed in the file list box. The user will then be able to see the selected picture or document or play selected media files. The user may also want to add some additional text as a remark to the file, such as why it is important to be embedded in a particular search term. This file will be stored in the database and will be loaded to the multitab pad in the search menu when the user clicks the quot;openquot; button. Otherwise, if no quick view or comments are listed on a document file, then a double click to a file name listed in the file list box will complete the same operation as clicking the "open39;39; button. The browsing windows will automatically unload whenever the embedding process is successfully implemented. Storing and loading a picture, an office document or a media file categorized as a binary data type is completed using a different process than other types of data such as text, integers, dates, etc. A binary data type needs a streaming process in order to store or to load a binary data type to and from a recordset. In order to do so, a "stream39;39; object that represents the binary text data needs to be constructed. Because this stream object is constructed for streaming the binary data, the stream type needs to be set as "adTypeBinary39;39;. The process of embedding a user39;s local document file can be described as follows. First,

a recordset will be opened and a new row of data will be added. Next, the chosen embedded file will be loaded to an open stream object using the "LoadFromFile39;39; method. To assign the binary data from the stream object to the field in the recordset, the "read39;39; method is used and then the stream object will be closed. Meanwhile, the streaming process to load the binary data from a recordset is slightly different. Before it can be loaded to a viewer in the file browsing window or multitab pad in the search menu, the binary data will be assigned to an open stream object by using the "write39;39; method and then will be saved into a temporary file by using the "savetofile39;39; method. This temporary file will actually be loaded into the viewer. Each of these temporary files are different in how they will be loaded and represented to the user. After the loading process of the viewer into a browsing window or multitab pad in the search menu has been completed, the temporary file (MyTempPictureFile or MyTempDocFile or MyTempMediaFile) will be deleted. The difference in how a user39;s embedded file loaded to search results side not depends on its type. A static image will be loaded into an image container, and an office file will be loaded into an OLE (object linked embedded) container. Meanwhile, moving pictures and sound will be played in a media file. In addition, since more than one picture, office or media file can be embedded into a single search term, a record locator is placed in each tab. This record locator can move from one record to the next or to the previous record and also allows jumping to the first or the last record. However, since most office or media files involve large bytes and therefore require more computer memory capacity, only the header of the record will be moved to the record locator. This file will only be loaded when the user clicks the "load39;39; command for the office file and "play39;39; button for the media file. Unlike a picture or media file that easily fits into a picture container or media player no matter how big the file, an office file cannot be properly viewed in a small OLE container. However, this technology allows the file to be opened in a computer program in which the file can be viewed properly. An example would include a Microsoft ® Office Word file that is embedded into a user39;s search term. This file would be opened in its original format in Microsoft ® Office Word whenever the user double-clicks the OLE container. This file will go back into the OLE container whenever the user closes Microsoft ® Office Word.

4. CONCLUSION

This paper briefed the concept of design and its related terminologies, especially the creative design process. It also explained the difficulties that have been faced by designers in order to collect information using the existing instruments. An application that works over the Internet search engines that fit with the way how a designer collecting information in a creative design process has been proposed and successfully implemented. Further research need to be addressed in order to reach a fully compliance information search tool for designers in creative design processes.