Leveraging Google Web Search Technology to Find Web-based Learning Objects

Cecilia Curlango-Rosas*, Gregorio A. Ponce†, Gabriel Lopez-Morteo*, Monica Mendiola*
*Universidad Autonoma de Baja California, Mexicali, Mexico
Email: curlango@uabc.mx, galopez@ing.mxl.uabc.mx
†San Diego State University, Calexico, USA
Email: gponce@mail.sdsu.edu

Abstract—The web contains objects that can be used to improve the learning process. However, the task of finding these objects is hampered by a lack of tools to help users focus on searching and find what they want. This article describes the Learning Object Search Tool Enhancer (LOBSTER) which assists users during the search process. Evidence is presented which suggests that LOBSTER helps users find what they are looking for and that they find it to be an easy to use and useful tool.

I. INTRODUCTION

The Internet provides access to a vast amount of material. Teachers and students must sift through it in order to find learning objects (LOs) to address their learning objectives. Some of this material was created with the intention that it be shared and with a clear formal learning objective in mind. LOs such as these are commonly housed in specialized repositories such as Merlot [1] and Ariadne [2]. These repositories provide their own search engines so users can search their collections. Other material consists of presentations and other learning aids that teachers publish on their personal and class web pages and is accessible to all. This material is also intentionally shared and has a formal learning objective. However, it is not indexed in a centralized database, so it is more difficult to find and external users, those who were not the original audience, must rely on general purpose search engines such as Google to find them. Another source of material is that which was not created with the purpose of being used in a classroom setting, but could be used for such a purpose. Such is the case of material published by various companies and organizations in the form of online manuals, white papers, tutorials and product specifications. To find this material, users again turn to search engines.

Finding LOs that are not stored in specialized repositories is not easy, in fact searching the web has been compared to searching for the proverbial needle in a haystack. One group of factors that contribute to this problem are those that are common to Internet searchers [3]. One of these is that users on average make few queries, on average 1.6 queries per search session. A second factor is the number of pages containing search results examined is small 2.35 pages containing 10 results each. A third factor is the make up of the search query. Search queries tend to have few terms, on average 2.21 terms each.

A second group of factors hindering the search for LOs are related to LOs themselves. Specialized portals are difficult to search through and some are accessible only to members of certain groups and consortia. In addition, these portals only contain a small amount of LOs, especially compared to the wealth of LOs available on the Internet. Also, there is no way to search all portals for available LOs [4]. This is especially true of portals which require that a username and password be provided before they can be accessed. In this manner, these portals actually remain hidden from web crawlers, which means that their contents do not make it into search engine indexes.

A third group of factors is related to language. The language that users specify their queries in directly impacts the number of results found returned by a search engine. Also, the language that users need their LOs to be in affects the number of LOs that they may find useful. Finally different search engines work best with certain languages [5].

In order to address the difficulties faced by teachers as they search the web to find suitable material for their classes, we propose a conceptual model of a learning object query and describe a tool we built that implements this model as well as provides assistance to the user. We further describe a study we performed in order to evaluate different aspects of our tool, such as whether it helps users find LOs and how it compares to searching through the Google user interface.

In this article we first present some of the literature regarding the search for LOs. Then we describe how we built our tool and explain the design considerations we made. In addition, we describe its user interface and how it assists searchers. We then present the results of our study and conclude with a summary of our contribution and a discussion of future areas we foresee exploring in order to improve searches.

II. RELATED WORK

Several efforts geared towards finding LOs have been documented. Farrell et al. [6] developed a search engine for LOs as part of a learning system used by employees at IBM. The search engine was able to find LOs from among a group of 500 LOs that were extracted by a group of experts and instructional designers from IBM Redbooks which are how-to books on IBM products. While this was a successful project, it would be difficult to extend to the web at large since the search engine requires that each LO have metadata that conform to the IEEE.
Learning Object Metadata standard and most LOs on the web do not.

Brosin and Vidal [7] propose tracking users as they interact with an online learning system. This information can then be fed to a search engine within the learning system so that LOs can be recommended based on the concepts and interests that users prefer. An advantage of this system is that users do not have to express their interests explicitly. A disadvantage is that the search engine only searches within its own LO repository. Ochoa and Duval [8] propose using Contextualized Attention Metadata, which is information regarding users' actions as they interact with tools, their context, and profile to make recommendations to users based on what they are doing. Our approach to searching for LOs is different from all those mentioned above because we consider the entire searchable web as our repository.

III. LOBSTER DESIGN AND IMPLEMENTATION

Our goal in designing LOBSTER was to assist users when they search for LOs from different disciplines. However, this first implementation is focused on a specific type of LO. Thus, we define a LO as a digital component viewed via the Web, that is used directly (as is), modified, or included in another LO, for the purpose of reaching a learning objective in the field of computer programming. The reason this definition is self-referencing is that a LO can be used as a component of another LO that is more complex. For example, one can include an Applet that simulates a sorting algorithm within a LO that is devoted to contrasting the efficiency of various sorting algorithms.

One of the design decisions that we made at an early stage of building LOBSTER was to use freely available components, libraries, and services whenever possible. This decision was based on a desire to develop LOBSTER as quickly as possible, limit the number of custom built components required, and reduce the cost of development.

In this spirit, we selected the Google Web Toolkit (GWT) [9] which provided us with several benefits. First, GWT freed us from having to address the differences that exist between web browsers, thus we were able to concentrate on the development process as a whole. Second, GWT provided us with an ample variety of components that we used to make LOBSTER’s user interface easy to use. In addition, we were able to use the Java programming language and all its stable and mature development tools which greatly eased the development process. Finally, by using GWT we left the door open to future expansion of LOBSTER’s features using an array of third party libraries that currently exist and are free for use.

We also integrated the Google Ajax Language API [10] into LOBSTER to provide users with translation assistance during query specification—a feature not commonly found in the popular search engines at this point of the search. While this API provides text translation between 41 languages, LOBSTER only uses it to translate user written terms to Spanish and English. These two languages were chosen based on the responses given by participants, who were computer science professors. When asked what languages they wrote search terms in, 23 indicated Spanish and English while 7 indicated Spanish only. Also, when asked what language they preferred the LOs to be in, 13 indicated Spanish only, while 17 indicated either language.

One issue we found with the Google Ajax Language API is that the translations provided by this API are not always appropriate for the context LOBSTER is focused on. For example when asked to translate the term arreglo, a common expression in computer programming, from Spanish to English, the API returns arrangements instead of arrays. To address this issue, LOBSTER provides the means for users to modify the translations to better describe the LO they are trying to find.

As we have stated previously, searches are performed using Google. This task is facilitated by two Google Ajax Search API components [11]: Web Search and Multimedia Search. The first component is used to search for LOs such as documents, slide presentations, and interactive LOs. The second component is used to search for Google for images and videos such as those found on YouTube. Besides the differences in the type of objects that these two components search for, they list search results differently. Web Search presents results that include text hyperlinks and short descriptions of the result along with a partial URL. Multimedia search for images includes the image, a two or three word text description, the size of the image and the URL. The search results for videos show: a still image from the video, a text hyperlink, a short description and either the YouTube or Google Video URL. By using the Google Ajax Search API we were able to easily leverage the power of the Google search engine.

IV. PROVIDING ASSISTANCE

We have identified two moments during the search process in which the user requires assistance: during the process of describing the desired LO and when examining the search results. This section details how individual elements in LOBSTER's user interface provide support to searchers during these two moments.

A. Query Specification Support

We conceive of a query for a LO as being composed of two major parts shown in Fig. 1: the Subject Matter and the Visual Aspect. Subject Matter is the component of a query which encapsulates the central learning objective of the LO. This can include both formally accepted names and descriptions as well as informally used but equally recognized terms and descriptions. In turn, the Visual Aspect of the query is composed of three subcomponents which address other aspects related to the physical appearance of the LO. The Format subcomponent refers to the LO's digital format or file type. The Language subcomponent indicates the preferred language for the LO, for example: English or Spanish. Finally the manner in which the LO organizes its content is indicated in the Structure subcomponent. By implementing this model
through LOBSTER's user interface we provide a guide that the user can follow to ensure that the description of the LO is as complete as possible which in turn is conducive to obtaining accurate search results.

Fig. 2 illustrates how LOBSTER implements the Subject Matter component in the QueryView. Since the LOs LOBSTER is focused on are from the field of computer programming, the Subject Matter component is broken up into two parts: Tema Principal del OA (main topic of LO) and Lenguaje de Programación (programming language). The motivation for having these two separate components is to obtain the central theme of the LO separately from the programming language since some topics in computer programming such as those related to algorithms or theoretical concepts can be viewed independently of a programming language. In this way LOBSTER facilitates obtaining a more fine-grained description of the LO's central topic.

LOBSTER assists the user in several ways when specifying the Subject Matter. First, assistance is provided through the automatic translation of terms. When the user finishes typing the terms in the main topic of LO, LOBSTER takes these terms and requests translations, to both English and Spanish, from the Google AJAX Language API. The resulting translations appear in the text boxes next to the corresponding flag buttons. The left side of Fig. 2 shows the resulting translations for the main topic of LO. The user can modify the translations by first pressing the flag button of the corresponding translation and then editing the entry.

Another way in which LOBSTER assists the user is by providing easy access to Google's Advanced Search without having to know the exact syntax of an advanced query. In the case of the main topic of LO, there is a check box at the bottom that reads Frase exacta (exact phrase). When the user selects this box, the terms are placed within quotation marks so that Google returns search results that include all the terms.

LOBSTER also provides assistance to the user as he/she writes the main topic of the LO and the programming language by suggesting possible terms based on what he/she types. We implemented this feature using the SuggestBox and MultiWordSuggestOracle components provided by GWT. Each MultiWordSuggestOracle contains the set of terms used by the SuggestBox to assist the user. The right side of Fig. 2 demonstrates how LOBSTER provides suggestions as the user types a programming language.

The Visual Aspect of the model is implemented in LOBSTER as shown in Fig. 3. The Format subcomponent in the model is implemented by the Formatos de OA (LO Formats) component in LOBSTER. This component lists supported LO formats and allows the user to select some or all of these. The formats are labeled according to the types of files they correspond to namely: (a) Documentos (documents), (b) Presentaciones (presentations), (c) Interactivo (interactive), (d) Imagen (image) and (e) Video. Selecting one of the first three formats causes a list of file types joined by multiple instances of the keyword OR to be added to the query. This list is expressed in Google's syntax for indicating file types, for example filetype:pdf for Portable Document Format. Selecting the documents format includes HTML files among the document types. However the HTML file type is treated differently from the other document types by triggering a search without specifying a file type to indicate regular web pages. Among the interactive formats, the Applet receives special treatment because it is not found as a stand-alone file, rather it is usually found embedded within web pages by using a special HTML tag. Thus for Applets only the term applet is added to the query to indicate this type of file. Selecting the last two formats, Image and Video, does not require adding additional terms in the form of file types to the query since each is directed to Google controls that specialize in retrieving images and videos.

The Language subcomponent in the model is implemented by the Idiomas del OA (LO Languages) component in LOBSTER. This component allows the user to easily select whether to search in English, Spanish or both languages. This component does not modify the query, rather it modifies the configuration of the Google search control to indicate the preferred language for search results.

LOBSTER takes the selections made in the LO Formats and LO Languages components and creates a number of independent SearchControllers corresponding to the cross product of the two components. Thus when all LO Formats and all LO Languages are selected LOBSTER creates a total of 14 specialized searches. Since all of these searches are made at the same time, it implies a time savings for the user since all that is required to initiate the multiple searches is a single query description.

The final subcomponent from the model, Structure, is implemented in LOBSTER by the Términos adicionales (Additional Terms) component. In terms of functionality, this component works in the same way as the main topic of LO component
V. Case Study

We conducted a two phased study to validate our approach to assisting users during searches. We wanted to know whether using LOBSTER to search led to a greater number of users finding LOs as compare to when they used Google directly, including both basic and advanced search. In addition, we wanted to know whether users found LOs faster with our approach than with the approach they usually use. Also of interest, was whether users considered LOBSTER a useful and easy to use tool for searching for LOs.

Our sample population consisted of a group of 30 university professors from the field of computer science who currently lecture at a university in northern Mexico. In the first phase we gathered data regarding how participants usually search the web to find LOs. In the second, participants searched for LOs using LOBSTER. This section details our study and presents its results.

A. Description of Study

1) Description of First Phase Activities: The first phase of our information gathering consisted of individual sessions with each of the study participants. An audio recording was made of each session. In addition all searches and interactions with the computer were recorded using video capture software. Participants were advised that they could use any browser they wished during the session and that they could download and install any software or file that they required during the searches. In short they were told to consider the computer on which they were working as if it were their own and to treat it as such. The only restriction was that they use Google for their searches.

At the start of each session, participants were informed of the purpose of the study and permission was obtained to record the session. Then participants were read three definitions of LOs including the one proposed by the IEEE's Learning Technology Standards Committee (LTSC) [13]. If they had questions regarding what a LO was, participants were able to obtain information from the person conducting the study.

For the first search, participants were given a scenario and then asked to describe, in as much detail as possible, the characteristics of the LO that they would like to find for the scenario. Once they had described the LO, they looked for it using Google. Participants could take as much time as they needed or wanted to look for the LO. Once they found it they would say so and stop the recording. They could also give up the search and stop the recording whenever they considered that further search would be pointless.

The first search had several objectives. One was to see whether participants would find exactly what they were looking for or settle for something else. Since we knew exactly what they were looking for, we could compare the LO they found with the description. In this way we could obtain a sense of what they considered to be a successful search. Another objective was to obtain data regarding the queries they used to search for the LOs, such as how descriptive the queries were, what language the queries were in, how many terms were used...
in each query. A third objective was to record the strategies that the participants used when searching for LOs. This included how many results they examined, how many pages of results they looked at, and whether they used the advanced search facilities provided by Google to enhance their queries.

Once participants concluded the first search, they were given a detailed description of another LO and asked to look it up. For this search, we selected an LO that we knew existed and could be found among the first 10 results of a Google search when a certain query was used. Again, participants could look for the LO until they either found it or gave up.

Once participants finished the second search activity they were asked to answer an on-line questionnaire. Some questions were general such as age, gender and number of hours they teach per week. Other questions were regarding the amount of time they spend looking for LOs, where they are when they do their searching, how often they find what they look for, and how they use the LOs they find to support their teaching. We also asked for a list of the search engines they normally use when looking for LOs.

We included a question in which participants were able to give us in essence a wish list of features they would like to see in a search engine that they believe would help them find LOs. The information from this question helped us to know what characteristics they felt were currently lacking in search engines so they could be included in LOBSTER’s design. In addition, this question gave us an idea of what useful features, which Google already provided, were actually unknown to our participants and should be emphasized with LOBSTER.

Another section of our questionnaire contained questions that are like those proposed in the Technology Acceptance Model (TAM). The TAM measures perceived ease of use and perceived usefulness [14]. We included these questions because we wanted to know how useful and easy to use participants found Google.

Finally, the questionnaire included questions regarding language. One of the questions asked what language they used to write the terms in the query and why. They were presented with the two more probable answers which are Spanish and English but were given the option of indicating another language. Another question asked for the language they preferred that the LOs be in. Again the most probable answers were given but the option to specify another was also provided.

2) Description of Second Phase Activities: The purpose of the second phase was to gather data regarding whether LOBSTER helped participants when they searched for LOs. This phase was also conducted in individual sessions with each of the study participants and took place approximately 6 months after the first phase. An audio recording was made of each session. In addition all activities that were performed on the computer were also recorded using a video capture software. Participants were advised that they could use any browser they wished during the session and that they could download and install any software or file that they required during the searches. They were asked to perform all of their searches using LOBSTER.

At the start of each session, participants were informed of the purpose of the session. Then they were read the definition of LO that focused on LOs from the field of computer programming. After this, participants were introduced to LOBSTER. They were given a verbal description of each of LOBSTER’s components and their features. Then they performed a guided search to exemplify LOBSTER’s functions and familiarize participants with the user interface. Once the demonstration concluded, participants were given a scenario for the first search and then asked to describe, in as much detail as possible, the characteristics of the LO that they would like to find for the scenario. The scenario was the same one used in the previous phase. Once they had described the LO, they looked for it using LOBSTER. Participants could take as much
time as they needed or wanted to look for the LO. Once they found it they would say so and stop the recording. They could also give up the search and stop the recording whenever they considered that further search would be pointless.

When participants concluded the first search, they were given a detailed description of another LO and asked to look for it. Again this LO was the same one they looked for during the first phase. Participants could look for the LO until they either found it or gave up. After concluding the second search, participants answered an online questionnaire that focused on how useful and easy to use they perceived LOBSTER to be.

B. Data processing

At the end of each phase, we coded the video recordings of participants’ search activities using an augmented version of the coding scheme proposed in [15]. In the coding process we reviewed each video and recorded information such as the terms used in a query, the position within the Google results page of search results examined, URL of the search results examined, the time spent examining each page, the actions performed in order to arrive at a page, and activities performed at each page. We added over forty codes to the original coding and classification scheme because the participants performed actions that had no code assigned to them, such as opening multiple tabs in a browser.

C. Results of Study

Results of our study are encouraging. The first LO that participants searched for in each session was one that they described for a scenario that was posed to them. That is, they searched for an LO of their own choosing, with most participants electing to search for a text-based explanation LO. Using Google’s interface, Basic and/or Advanced, 24 (80%) participants were able to successfully find an LO that met their requirements as compared to 29 (96%) successful participants when using LOBSTER. We applied a test of significance of two proportions and found that this difference is statistically significant ($p < 0.037$) at a confidence interval of 95%.

The second LO that participants searched for was also the same for each session. They were asked to find a LO that demonstrated in a dynamic and visual manner how a particular search algorithm worked. For this LO, 12 (40%) found it using Google, while 20 (66%) found it using LOBSTER: the difference is also statistically significant ($p < 0.032$) at a confidence interval of 95%.

Table I shows the time in minutes that participants required to look for the LOs in both rounds of searches. The table shows first, second and third quartiles (Q1, Median, Q3) as well as minimum and maximum times. Due to technical problems with the video capture software, the recordings for two of the participants in the second round were lost. Therefore, these results correspond to 28 participants. We reiterate that there was no restriction on the amount of time that participants could spend on their searches as they were instructed to search for the LOs until they found what they were looking for or until they decided to stop.

<table>
<thead>
<tr>
<th>Ease of use</th>
<th>Search Tool</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>5.089</td>
<td>1.095</td>
<td></td>
</tr>
<tr>
<td>LOBSTER</td>
<td>6.539</td>
<td>0.471</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Usefulness</th>
<th>Search Tool</th>
<th>Mean</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google</td>
<td>5.433</td>
<td>0.820</td>
<td></td>
</tr>
<tr>
<td>LOBSTER</td>
<td>6.217</td>
<td>0.578</td>
<td></td>
</tr>
</tbody>
</table>

Using LOBSTER, less time was required to search for the first LO. The median time decreased from 8.5 minutes with Google to 3.0 minutes with LOBSTER. A similar decrease occurred during the search for the second LO. The median time decreased from 33.5 minutes with Google to 26.5 minutes with LOBSTER. Interestingly, both minimum and maximum times increased using LOBSTER for the second LO. Out of the 5 participants that spent the longest time using LOBSTER, only 1 found the second LO where as before none had found it. This may indicate that users were willing to spend more time exploring search results with LOBSTER’s interface even when they did not immediately find what they were looking for on the Internet.

After applying a logarithmic transformation to the data to address skewness and outliers we used a matched-pair t-test to determine if there was a significant difference in mean search times for both tasks. The analysis reveals a statistical difference ($p < 0.022$) in the mean time (8.39 minutes) to successfully complete the first task using LOBSTER as compared to Google (11.93 minutes). However, on the second task, there was no significant difference ($p < 0.413$) between LOBSTER (33.11 minutes) and Google (35.83 minutes).

D. User Perceptions

The Technology Acceptance Model (TAM) developed by Davis [14] predicts users’ acceptance of software. The TAM provides a 12 item instrument that measures how easy to use and useful users perceive a piece of software. Davis defines perceived usefulness as the degree to which a person believes that using a particular system would enhance his or her job performance. While ease of use is defined as the degree to which a person believes that using a particular system would be free of effort. He further reports that though both aspects are predictors of system adoption, there is a stronger correlation between usefulness and system adoption than between ease of use and system adoption.

During our study we used the instrument provided by the TAM to obtain participants perceptions of Google and LOBSTER’s ease of use and usefulness. Each question asks participants to rate the extent to which they agree with the statement presented by each item. Statements are rated on a 7 item Likert scale where responses vary from 1 completely disagree to 7 completely agree. Table II shows the mean and standard deviation of the results we obtained for participants perceptions regarding Google and LOBSTER.
TABLE I
LO SEARCH TIMES IN MINUTES

<table>
<thead>
<tr>
<th>Task 1</th>
<th>Session</th>
<th>Min.*</th>
<th>Q1*</th>
<th>Median*</th>
<th>Q3*</th>
<th>Max.*</th>
<th>Mean*</th>
<th>Std. Dev.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unassisted Search - Google</td>
<td>0</td>
<td>4</td>
<td>8.5</td>
<td>18</td>
<td>47</td>
<td>11.93</td>
<td>10.64</td>
<td></td>
</tr>
<tr>
<td>Assisted Search - LOBSTER</td>
<td>0</td>
<td>2</td>
<td>3.0</td>
<td>11</td>
<td>42</td>
<td>8.39</td>
<td>11.12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task 2</th>
<th>Session</th>
<th>Min.*</th>
<th>Q1*</th>
<th>Median*</th>
<th>Q3*</th>
<th>Max.*</th>
<th>Mean*</th>
<th>Std. Dev.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unassisted Search - Google</td>
<td>1</td>
<td>21</td>
<td>33.5</td>
<td>51</td>
<td>94</td>
<td>35.83</td>
<td>20.81</td>
<td></td>
</tr>
<tr>
<td>Assisted Search - LOBSTER</td>
<td>6</td>
<td>9.75</td>
<td>26.5</td>
<td>42.75</td>
<td>103</td>
<td>33.11</td>
<td>27.62</td>
<td></td>
</tr>
</tbody>
</table>

*Note: Values given in minutes.

Again we used a matched-pair t-test to determine if there was a significant difference in mean ease of use for both search tools. The analysis reveals a statistical difference (p < .000) in ease of use for LOBSTER (6.539) as compared to Google (5.089). There was also a statistical difference (p < .000) between usefulness for LOBSTER (6.217) and Google (5.433). Participants comments during the sessions in the second phase confirmed that they found LOBSTER easier to use and more useful than Google's interface.

Participants were also given the opportunity of expressing their views on LOBSTER using their own words. One of the participants said, "It provides a good guide by filtering the information one wants to find." Another expressed, "You can make a search with greater precision." A third participant said, "I don't consider myself a good researcher, but it (LOBSTER) is easy to manipulate and after using it for some time I believe I would be able to make faster and more precise searches." As can be seen from this selection of comments, users view LOBSTER in a positive light and their comments agree with the answers to the TAM questionnaire.

VI. CONCLUSIONS AND FUTURE WORK

Some approaches to finding LOs on the web restrict the LOs in some manner such that it is difficult to apply the same approach with LOs in general. Our approach differs from others in several ways. First we strive to be open to different types of LOs. While our test application, LOBSTER, recognizes a specific group of file formats, its architecture is flexible enough to accommodate more or different formats. Second, the model of a LO query appears to be applicable to other fields of knowledge with slight modifications. LOBSTER, which implements the LO query model is adaptable to other domains with some changes to labels and the MultiWordSuggestOracle terms, such that these reflect the appropriate vocabulary for the specific field of knowledge. In fact, we have anecdotal evidence that professors from fields other than computer programming have used LOBSTER successfully to find LOs.

By eliciting as complete a description of the LO as possible from the user, LOBSTER assists users in structuring queries that produce better search results as compared to having the user structure his/her queries without any assistance. LOBSTER goes beyond being a mere query builder since it not only assembles a query based on the user's description of the LO but it in fact launches up to 14 searches simultaneously and presents the users with clear, organized listings with 8 results each. In order to perform the same number of searches directly via Google, the user would have to: (a) manually build 7 complex queries using terms in English and another 7 complex queries using terms in Spanish, (b) load Google web search on 10 separate browsers windows or tabs, (c) load Google image search on 2 separate browsers windows or tabs, (d) load Google video search on 2 separate browsers windows or tabs, (e) configure the Google interfaces so that half return results in English and the other half in Spanish, (f) perform the 14 searches via each one of the 14 separate Google interfaces, and (g) examine each browser window or tab listing the Google search results. These steps must be repeated each time there is even a small change in the query terms.

LOBSTER's user friendly interface allows searchers to modify their queries easily. By preserving the query view, LOBSTER minimizes the amount of information that the user must remember between queries. Finally, enhancements such as automatic query translation and multi language queries, set LOBSTER's user interface apart from others. These and other features contribute to users' perception that LOBSTER's user interface is easier to use and more useful than Google's.

The conceptual model of a learning object query and its implementation within LOBSTER is applicable when working with other metadata such as that used by other search engines, for example Yahoo! as well as with Learning Object Metadata (LOM) sponsored by IEEE's LTSC. In fact, adapting LOBSTER to work with LOM and integrating it within existing LO repositories, would allow users to search the LO repositories as well as the web.

In the future, we will explore the integration of social networking into LOBSTER to further assist searchers by providing access to community based recommendations of LOs.

ACKNOWLEDGMENT

The authors would like acknowledge the support provided by the Universidad Autónoma de Baja California, particularly its School of Engineering for making this work possible.

REFERENCES