Semantic search of tools for collaborative learning with the Ontoolsearch system

Guillermo Vega-Gorgojo a,*, Miguel L. Bote-Lorenzo a, Juan I. Asensio-Pérez a, Eduardo Gómez-Sánchez a, Yannis A. Dimitriadis a, Iván M. Jorrín-Abellán b

aSchool of Telecommunications Engineering, University of Valladolid, Camino del Cementerio s/n, 47011, Valladolid, Spain
bFaculty of Education, University of Valladolid, Paseo de Belén 1, 47011, Valladolid, Spain

Abstract

This paper introduces Ontoolsearch, a new search system that can be employed by educators in order to find suitable tools for supporting collaborative learning settings. Current tool search facilities commonly allow simple keyword searches, limiting the accuracy of obtained results. In contrast, Ontoolsearch supports semantic querying of tool knowledge bases annotated with the Ontoolcole ontology, specifically designed to fit educators’ questions. Moreover, Ontoolsearch offers an innovative direct manipulation interface to educators, intended to facilitate query formulation as well as the analysis of obtained results. To evaluate this proposal, a group of educators was engaged in a formal comparison study of Ontoolsearch with a keyword search facility based on Lucene. Participants had to find tools for six search tasks based on real CSCL settings, using both systems alternatively. Evaluation results showed that retrieval performance was significantly better with Ontoolsearch, despite educators’ previous experience with keyword searches. Further, educators rated very positively the user interface of Ontoolsearch and considered this system very useful to find tools for their own learning situations.

Key words: Collaborative learning, search process, graphical user interfaces, user-centered design, ontology languages

* Corresponding author. Tel: +34 983423000
Email addresses: guiveg@tel.uva.es (Guillermo Vega-Gorgojo), migbot@tel.uva.es (Miguel L. Bote-Lorenzo), juaase@tel.uva.es (Juan I. Asensio-Pérez), edugom@tel.uva.es (Eduardo Gómez-Sánchez), yannis@tel.uva.es (Yannis A. Dimitriadis), ivanco@doe.uva.es (Iván M. Jorrín-Abellán).
1 Introduction

With the advent of Information and Communication Technologies (ICTs), different technological approaches have been proposed to support learning. One major trend makes use of ICTs to facilitate the accumulation, organization and delivery of content [13]. Many web-based training systems and learning platforms promote this model based on traditional pedagogies that emphasize knowledge transmission. In contrast, socio-constructivist learning theories [17] reflect the importance of social processes as an essential element of learning and advocate that knowledge cannot be simply transmitted but constructed from our own experience. Computer Supported Collaborative Learning (CSCL) [23] is a research field with roots in socio-constructivism focused on the use of ICTs to facilitate the building of new knowledge by means of the interactions that take place within the members of a group.

A CSCL situation typically consists of a sequence of both individual and collaborative activities that should be performed by learners working with computers in order to achieve predefined pedagogical objectives. To support a CSCL situation, different types of tools can be employed, even not specifically designed for learning nor for collaboration [30]. For instance, a pair of students can collaborate around the computer running a simulation of a computer network and then have a face-to-face discussion of the obtained results. Moreover, not all of the activities involved in a CSCL situation need to be collaborative, e.g. after the previous activity, each student can write an individual report using an off-the-self word processor. Nevertheless, collaborative technology can be effectively employed in many cases: for example, a pair of students can join a shared visualization of a network simulation with the Collaborative Network Simulation Environment (CNSE) [9] and then cowrite a report employing a synchronous collaborative text editor.

There is a growing demand for tools to facilitate the realization of different CSCL situations due to the diversity of curricula, teaching styles and cultural differences among institutions, or even among educators at the same institution [42,30,43]. Considering this situation, educators have to carefully select appropriate tools to support their intended learning scenarios, so tool cataloguing and searching should be facilitated as much as possible. Therefore, proper tool descriptions, i.e. tool metadata, should be provided to enable the development of search facilities targeted to educators’ needs. However, most efforts on learning metadata have been devoted on the sharing and cataloguing of so-called learning objects [31], i.e. specific chunks of educational contents, due to the prominence of web-based training systems and content-oriented educational platforms.

As a result, there are sparse tool search facilities available for educators, such
as the e-Learning Centre [28], [2] or the EduTech wiki [54]. Simple keyword matching techniques are commonly employed in existing search facilities, although keyword searches are prone to obtain unexpected responses since a term may have more than one distinct meaning [27,36,58]. Moreover, relevant results may not be retrieved because the terms employed to describe them do not match the query keywords, even though their meanings are similar [12, pp. 91-95]. Given these limitations, commonly known as “synonymy” and “polysemy”, there are ongoing proposals to enable semantic searches, specially in the context of the Semantic Web [7]. A semantic search denotes a concept about which the user is trying to gather information [15]. This way, it is possible to develop sophisticated search facilities that understand the semantics of a query in order to properly respond. To achieve it, ontologies [14] can be used to explicitly formalize a conceptual domain.

In their previous work, the authors have proposed the Ontoolcole ontology for the annotation of tools enabling the formulation of semantic searches [57,56]. Briefly, Ontoolcole aims at describing tools that can be employed to support CSCL settings using comprehensible abstractions for educators referred to the offered functionality of a tool. Results of a preliminary study with educators showed that this ontology can express their information needs for tools in the context of their real practice [56]. However, a search system is still required in order to fully support educators in the process of searching tools for the tasks to be performed by learners.

The user interface of such a system is specially challenging, since it should allow educators to understand their information needs and express them into queries. One common approach that can be found in semantic-based search systems is the use of formal query languages such as SPARQL [38]. Although powerful and flexible, these languages have a complex syntax that has to be learned by end users. Indeed, this interaction style does not seem to be suitable for educators since they often lack technical skills [46] and it cannot be assumed that they will invest the required time to learn a complex query language. Moreover, educators are expected to search tools sporadically, pushing to simpler and easier to learn interfaces. Besides, the search system should facilitate the comprehension of the underlying information structure, allowing educators to translate their information needs into queries.

This paper proposes a system named Ontoolsearch that is capable of processing semantic searches of tools annotated with the Ontoolcole ontology. In order to meet educators’ needs, the user interface has been devised following a participatory design strategy [33] involving both final users and software developers. Thus, Ontoolsearch offers a direct manipulation graphical interface to educators with the aim of facilitating the formulation of their intended questions. The paper also compares this approach with keyword-based searches, which are sometimes preferred by end users [52]. Indeed, keyword search based on tra-
ditional information retrieval techniques [4] is the dominant search paradigm employed in many popular applications such as web search engines. To perform this comparison, a group of educators were engaged to carry out some predefined search tasks founded on authentic CSCL scenarios using alternatively both Ontoolsearch and a keyword search facility based on the popular text search engine Lucene [1]. The main goal of the study was to assess with educators whether semantic searches with Ontoolsearch could be preferable to conventional keyword-based searches in this context, as well as to identify the strengths, weaknesses and significance of this new system.

The rest of this paper is organized as follows: Section 2 succinctly describes the Ontoolcole ontology that was originally proposed by the authors to annotate tools for CSCL employing meaningful abstractions for educators. The Ontoolsearch system is presented in Section 3, while Section 4 reports the evaluation of the system that has been carried out. Finally, conclusions and future work are given in Section 5.

2 Ontoolcole: an ontology of tools for collaborative learning

Ontoolcole [57,56] is an evolving ontology of tools for collaborative learning that was designed with the aim of supporting educators in the search of such tools. This ontology was developed following the well-known On-To-Knowledge methodology [47] which specifies an iterative ontology development process targeted to the final application that will use it. In order to acquire the necessary knowledge that Ontoolcole should include, different input sources were considered. Specifically, a competency questions study was carried out with a group of educators that were asked to pose questions for searching tools referred to their own CSCL settings. The analysis of these questions was then complemented with a literature review of tools employed in CSCL, serving to obtain the core elements that the ontology should include.

After the knowledge acquisition phase, the conceptualization of Ontoolcole was accomplished. Fig. 1 represents the basic conceptual model that contains the most important concepts and their relations. Specifically, the functionality of a Tool is defined by the Tasks that it can support, performed by an Actor maybe playing a specific Role. The realization of a Task may require an Artifact as input and/or produce an Artifact as output. These concepts have been further refined into other more specialized ones in order to allow the description of a wide range of tools for collaborative learning. Thus, the Actor concept is specialized to Person, Group and ComputerSystem representing the possible actors that may perform a task. Similarly, five prototypical task types, inspired in [23], were introduced to represent the main uses of a tool in a learning context: Perception, Construction, Communication, Computation and
Ontoolcole also includes a taxonomy of artifact types in order to accommodate the different products that may be used or created during the realization of a task. The well-known MIME types [19] have been employed to define the format of an artifact, allowing to specify that a particular instance is formatted in a specific MIME type such as application/pdf. Moreover, it can be specified that a tool can keep archival storage using the property providesStorage (see Fig. 1). Ontoolcole also incorporates a predefined set of tool definitions corresponding to well-known tool types, e.g. “concept map tool: any tool that allows a person or a group to model a concept map”. Combining all these elements, the capabilities of a specific tool can be specified, while educators can perform their intended searches either using tool types or supported task types as a basis.

Ontoolcole has been formalized in Description Logics [3] and implemented in the Semantic Web language OWL DL [6]. A wide range of tools have been annotated with this ontology, such as different instances of document repositories, drawing tools, collaborative text editors or multimedia players among others. For instance, the following RDF snippet shows the description of Imagination Cubed, a tool that allows a group to synchronously draw images as well as to exchange text messages:

```
<rdf:Description rdf:about="#imaginationcubed">
  <rdf:type rdf:resource="#Tool"/>
  <hasName rdf:datatype="string">Imagination Cubed</hasName>
  <hasURL rdf:datatype="string">http://www.imaginationcubed.com/</hasURL>
  <hasDescription rdf:datatype="string">Imagination cubed is a neat web-based networked whiteboard that allows multiple users to draw concurrently</hasDescription>
  <supportsTask rdf:resource="#icDrawing"/>
  <supportsTask rdf:resource="#icCommunication"/>
</rdf:Description>
```

```
<rdf:Description rdf:about="#icDrawing">
  <rdf:type rdf:resource="#Drawing"/>
  <hasTimeOfInteraction rdf:datatype="string">synchronous</hasTimeOfInteraction>
  <isPerformedBy rdf:resource="#toolGroup"/>
</rdf:Description>
```
In order to assess whether Ontoolcole can support educators’ questions for the search of tools, a preliminary evaluation was carried out. Briefly, a group of educators described several educational settings based on their real practice and posed a set of questions to obtain tools to support them. These questions were translated into Ontoolcole-compliant queries, using the elements defined in this ontology. Translated queries were then paraphrased back to natural text and returned to participants who were asked to rate and provide feedback about the translation quality. Although questions were not restricted at all, evaluation results showed that Ontoolcole abstractions fit educators’ questions. More details of this preliminary evaluation, as well as of Ontoolcole, can be obtained in [56].

3 Ontoolsearch: a system for searching tools for collaborative learning

Once the Ontoolcole ontology has been developed, it can be employed to describe tools thus enabling their potential search. The envisioned usage scenario of this proposal is graphically depicted in Fig. 2. Providers such as educational institutions or software enterprises can offer their tools to the CSCL community. They will create Ontoolcole-based tool annotations either with a generic ontology editor like Protégé [22] or with a specialized annotation tool targeted to Ontoolcole. Produced descriptions will be published in the Web to make them widely accessible. A Semantic Web crawler such as Swoogle [11] can then be instructed to gather tool annotations. Retrieved tool metadata will be employed to populate the knowledge base of Ontoolsearch. Finally, educators willing to find tools for their CSCL settings can then query the search system.

For the successful realization of this scenario, it is necessary to provide an appropriate search system to educators. Such a system should guide them through the search process in order to obtain suitable tools for their information needs. With this aim, the authors have developed the Ontoolsearch system that is the topic of this section. First, the main design criteria will be discussed. Next, the logical architecture of Ontoolsearch are presented. Then, a
Figure 2. General usage scenario of Ontoolcole/Ontoolsearch.

depiction of the proposed user interface follows. Finally, some implementation remarks are given regarding the developed Ontoolsearch prototype.

3.1 Design criteria of Ontoolsearch

Ontoolsearch is a system devised for educators in order to assist them in the search of tools annotated in conformance to the Ontoolcole ontology. As most information retrieval systems, Ontoolsearch has two main functionalities: the specification of queries and the presentation of obtained results. More specifically, Ontoolsearch should help educators in the process of formulating their information needs as queries. Then, the system will process these queries and present the retrieved tools in an appropriate way to educators, facilitating the analysis of results. Throughout this subsection, the criteria that have guided the design of Ontoolsearch will be discussed, stressing the key principles in bold font.

Since the tool metadata are formalized in a machine-processable knowledge base, they can be exploited by Ontoolsearch to support the formulation of semantic searches referred to the concepts defined in Ontoolcole. It is commonly agreed that better retrieval performance can be achieved with semantic searches than with traditional keyword-based searches [8]. Retrieval performance is typically assessed with the recall and precision measures [4, pp. 73-79]; recall accounts the fraction of all relevant responses that are returned by the search, while precision measures the fraction of the retrieved responses that are relevant to the search. Since a semantic search does not suffer from the synonymity problem of keywords and hierarchical relations of concepts can be exploited, better recall can be achieved than with keyword-based searches. Further, a semantic search can increase precision by allowing more structured queries and removing the ambiguity of keywords, i.e. term polysemy.
Despite these expected benefits, one challenge refers to the support of educators in the construction of ontology-based queries. Since query specification is a general issue in semantic-based retrieval systems, different approaches can be found in the literature. One trend is the use of formal query languages such as SPARQL [38]. These languages attempt to be sufficiently expressive in order to allow the formulation of powerful and flexible queries. However, their complex syntax may discourage educators to use the system, so other simpler and easier to learn interaction styles should be considered. For instance, this SPARQL query requests tools that can be employed by groups to synchronously draw an image:

```
SELECT ?toolname
WHERE {
  ?tool a :Tool .
  ?task a :Drawing .
  ?actor a :Group .
  ?tool :hasName ?toolname .
  ?task :isPerformedBy ?actor .
  ?task :hasTimeOfInteraction "synchronous" .
}
```

Another option is to offer form fill-in and menu selection interfaces so that the user no longer has to learn a complex query syntax. For instance, KIM [37] is a knowledge and information management infrastructure allowing annotation, indexing, and retrieval of documents with respect to real world concepts. KIM offers a form fill-in interface for posing semantic queries. In this manner, query construction is simplified although expressiveness is limited, e.g. a maximum of three concepts can be employed in a query with KIM. Besides, drop-down menus could be less effective when a large set of concepts is defined in the knowledge base, as it is the case of Ontoolcole; for example, there are 46 tool types defined in this ontology.

Another common approach is to provide a simple entry form to query semantic data [18]. One advantage is that little knowledge of the system and the underlying data structure is required to construct a query. Indeed, this user interface is typically employed in information retrieval systems such as web searchers. However, user keywords have to be transformed into ontology-based queries using techniques such as spread activation [41] or thesauri exploitation [10], thus involving significant complexity. While this approach offers a general-purpose solution for querying different datasets of metadata, it does not give clues about the information structure to users. Due to this, specialized interfaces providing the necessary context for a search seem more adequate in specific domains as CSCL.

Therefore, the decision in Ontoolsearch has been to offer a **graphical direct manipulation interface**. This type of interfaces is commonly easier to use than other methods [45, ch. 6] due to: (1) continuous representation of the object of interest, (2) physical actions or button presses instead of complex
syntax, and (3) rapid incremental reversible operations whose impact on the object of interest is immediately visible. Moreover, direct manipulation interfaces often evoke enthusiasm from users, facilitating the learnability of the system [16]. These interfaces are not free of drawbacks though, since coding can involve significant complexity and meaningful visual elements for users are required [45, ch. 6]. In order to lessen the latter effect, a participatory design strategy has been followed to design the user interface with both software developers and educators. Participatory design is a common methodology to involve stakeholders to identify requirements and give feedback of preliminary prototypes [33].

Thus, several interface prototypes were iteratively proposed to educators until agreeing the final design. One of its key features was to allow the formulation of queries either using tool types or supported task types, since the Ontoolcole ontology was designed to provide these two complementary views (see section 2). Another issue of the user interface of Ontoolsearch consisted in providing an adequate representation of the taxonomies of tool types and task types to educators. Tree structures and graphs were considered to fulfil this goal since they are commonly employed to visualize taxonomies. A graph representation was chosen since it is more adequate to visualize multiple inheritance relations and makes better usage of screen space than tree structures.

3.2 Logical architecture of Ontoolsearch

The logical architecture of Ontoolsearch consists of a set of components, shown in Fig. 3, that are depicted next in the context of a typical information access process based on [44]. The educator begins with a preliminary idea of the tool she requires to support her CSCL setting (event 1 in Fig. 3). Interacting with the search user interface she can scan the information structure and acknowledge available query options. Then, she can begin to formulate her intended question manipulating the elements of the search interface (2). During this phase, each change of the ongoing query is sent to the search manager (3). This component maintains the state of the user query, listening to modification events sent by the search interface. Eventually, the educator can request to start the search through the search user interface component (4).

At this step, the educator waits till the system obtains the results to be presented. Initially, the search interface notifies the user request to the search manager (5). This component prepares a processable query for the search engine (6) that is in charge of retrieving those tools that comply with the user query. This is accomplished checking the tool knowledge base composed of Ontoolcole-based tool annotations. Thus, found tools are sent back to the
search manager (7) and then forwarded to the results user interface (8). This module displays these results, allowing the educator to evaluate their suitability (9). After completing this process, the educator can refine her query and begin a new cycle until finding a tool that satisfies her information need.

Note that the proposed architecture stresses the separation of concerns among the components of Ontoolsearch. Indeed, the well-known model-view-controller (MVC) pattern [24] has been applied in order to decouple the business logic from user interface considerations. Specifically, the search manager maintains the model, corresponding to the user query and the set of results. The user interface renders the model and embeds the controller that notifies the search manager of user actions resulting in a change in the state of the model. This way, it is possible to modify the visual appearance of the user interface without affecting the other components of the system. Moreover, the search engine does not require to know the structure of the Ontoolcole ontology since the search manager component is responsible of producing processable queries. Thus, an off-the-self search engine can be employed given that it understands the formalism of the tool knowledge base.

3.3 User interface of Ontoolsearch

The design of the user interface of the system is of great importance in order to allow educators to express their information needs into ontology-based queries and to support the analysis of obtained results. As discussed above, a design decision was to employ a direct manipulation interface for Ontoolsearch. With this approach, the aim is to simplify the process of query construction, hiding the formalism and the implementation details of the underlying ontology to educators. Moreover, the representation of visual elements in the user interface should allow the rapid communication of the abstractions modeled in Ontoolcole, since humans are highly attuned to images and visual information [26].

Concerning the search interface, the main challenge was to define which type of queries could be composed. A key requirement was to employ tool and task types as a basis since these are the core elements demanded by educators to
express their information needs (see subsection 3.1). Thus, it was decided to allow the formulation of queries expressed as a list of tasks to be supported, including the possible actor types as well as the input and output artifacts. Asking for a specific tool type was managed presenting the set of the supported tasks obtained from the ontology tool definition that can be incorporated to the user query. Restrictions can then be added to refine a query, such as changing an actor type, specifying an asynchronous group task or defining the MIME type of an artifact.

After deciding which query types should be supported, the search interface of Ontoolsearch was designed. Fig. 4 shows a snapshot of this interface during a query formulation process. The upper left frame is intended to provide educators with a good starting point for a search, displaying the graph of tool or task types that can be switched from one view to the other. The graph can be manipulated by moving, expanding, collapsing and hiding nodes, allowing educators to control the information to display. When a node is selected, the upper right frame displays a visual representation of supported tasks, along with the actors and artifacts, as shown in Fig. 4 for the whiteboard tool type. These mechanisms in conjunction allow educators to scan the information structure facilitating its comprehension.
Figure 5. Snapshot of the results interface of Ontoolsearch in which several tools were retrieved that comply to the query shown in Fig. 4.

The ongoing query is shown in the bottom left frame as a list of actor-task-artifact rows. So, when the educator explores the tool or the task graph, she can eventually select a node and include it in the query. It can then be refined by specifying additional restrictions, such as defining the MIME type of an artifact, using the controls enabled in the lower right frame. When the educator is done with her query, she can request to process the search.

Retrieved tool instances are then presented through the results interface. Fig. 5 shows a snapshot of this interface; the upper part displays a summary of the submitted query while the set of obtained tools are listed alphabetically in the left. When an item of this list is selected, a description of the tool is displayed along with the tool categories and the tool website. In addition, the lower right part shows a graph with the selected tool and the categories to which it belongs. The educator can then go back to refine her query, start a new one or access the session history to restore a previous query or its results. It is expected that during this process, the educator will eventually find some tools that satisfy her information needs.
3.4 Ontoolsearch prototype

A prototype of Ontoolsearch has been already implemented in Java in order to reuse available software libraries coded in this language. Specifically, the Swing Application Programming Interface (API) [50] has been employed to develop the graphical components of the user interface, such as buttons or pop-up windows. Although there is no support in Swing for the visualization and manipulation of graphs, the open-source Java Universal Network/Graph (JUNG) framework [35] was easily adapted for Ontoolsearch. Besides, the search user interface needs to inspect the Ontoolcole ontology in order to display appropriate representations and controls, e.g. showing a group icon to indicate which actor can perform a communication task. The Protégé-OWL API [48] has been employed to analyze Ontoolcole, simplifying the development of the search interface.

For handling the ongoing query a new OWL concept is created on the fly by the search manager. Each time the educator modifies the state of the query through the search interface, the search manager is notified and reacts updating the constraints of the associated OWL concept. For the processing of queries it was decided to employ an existing search engine to reduce development efforts. Indeed, an off-the-self DL reasoner can fulfil this task since the tool knowledge base is implemented in OWL DL and it can be translated into a Description Logics representation. Thus, the search manager has to transmit the query to be processed to a DL reasoner. Fortunately, standardization efforts have proposed a specification for connecting to DL reasoners. The DL Implementation Group (DIG) interface [5] defines constructs to submit a knowledge base to a DL reasoner and to access to inference services such as instance retrieval. Since the Protégé-OWL API provides a DIG implementation to access an external DIG-compliant reasoner, this specific API has been employed in the prototype of Ontoolsearch to support the submission of queries. Further, RacerPro [39] has been chosen as the DL reasoner due to its compliance with the DIG specification. This way, the prototype is not tied to a specific reasoner, enabling its substitution by another one that adheres to the DIG interface.

Since DL reasoners are computationally demanding, current implementation of Ontoolsearch follows the well-known client-server architecture model. Offering a light-weight client enables its installation in a wide range of machines, while a dedicated server can provide the necessary resources to respond to user queries. On the one hand, the server part incorporates a RacerPro installation attached with the knowledge base and a proxy to handle the communication with Ontoolsearch clients. On the other hand, the client bundle includes the user interface and the search manager functionalities. Java Web Start technology [51] is employed to deploy Ontoolsearch clients in order to facilitate
its accessibility to educators. When a client is launched, it requests the tool and task graphs to construct the search interface, as well as the Ontoolcole ontology; these files are then served by the proxy. Afterwards, the client can submit queries to the proxy and retrieve obtained responses. This client-server communication is supported with the Java Remote Method Invocation (RMI) [49], since RMI offers a simple and mature mechanism to communicate distributed Java objects. This prototype is already available to the community for testing purposes and can be accessed from the Ontoolsearch website [55].

4 Evaluation

This section describes the study that has been carried out in order to evaluate the suitability of Ontoolsearch for the CSCL domain. The experimental design of this study is inspired in the interactive track of the Text REtrieval Conferences (TREC) [25] which has formalized an evaluation method of systems designed to support interactive information retrieval that has become the de facto standard [21]. Briefly, a typical evaluation experiment involves a group of searchers that use one or more experimental systems to find information about a small set of prescribed search tasks. A control system is also employed as a baseline for comparison purposes. Common outcome measures are performance and usability; performance measures are based on the number of relevant documents searchers find, while usability measures are based on searchers’ responses to questionnaires or interactions with the system.

Since keyword search is the dominant approach for information seeking, the high-level goal of the evaluation was to assess with educators whether semantic searches with Ontoolsearch could be preferable to conventional keyword searches when seeking tools to support CSCL settings. Similarly to the TREC interactive track, the study was designed as a formal comparison of the experimental system Ontoolsearch and the control system Regain [40]. Regain is a typical keyword-based search facility based on the well-known text search engine Lucene [1], offering a Google-like user interface. Thus, a group of educators volunteered to perform a set of predefined search tasks using alternatively Ontoolsearch and Regain.

In order to evaluate which system could be preferable in this context, a more specific goal was to assess the retrieval performance based on the relevant tools found with each system. Besides the outcome, system logs were collected to examine how educators use these systems to find suitable tools for CSCL settings. Finally, it was desired to analyze system usability as well as to explore the strengths and weaknesses of Ontoolsearch, so user feedback was collected. For the evaluation of results, a mixed methods approach [53] has been followed analyzing both quantitative and qualitative data sources. These methods are
commonly employed to better comprehend the study under evaluation since they can provide additional information that enable further interpretation of the quantitative data. Indeed, they are specially suited for the analysis of human processes that are inherently complex [32].

4.1 Experimental setup

For this study, a testbed of 100 tools was annotated with the Ontoolcole ontology and employed as input of the Ontoolsearch system. This testbed included a wide range of tool types: 17 processing tools (e.g. compilers, simulators), 23 information management tools (e.g. document repositories, bulletin boards), 68 construction tools (e.g. drawing tools, slide composers), 58 representation tools (e.g. audio players, image viewers) and 17 communication tools (e.g. chats, videoconference tools). The complete knowledge base includes 466 classes, 38 properties and 1192 instances, taking 656 KB in OWL format. Correspondingly, a document set was prepared for the Regain system, containing a textual description for each tool annotated with Ontoolcole. These descriptions referred to the functionality of the tools and were constructed combining the information available on the Wikipedia and each tool website. Specifically, the Wikipedia was employed to provide a brief and independent overview of each tool. For this collection of 100 documents, the median number of terms per document is 98.5 and the mean is 107.7.

Then, a set of six search tasks was chosen for the study. They were intended to be representative of real CSCL settings requiring tools to support them. To accomplish this requirement, the search tasks were selected from the pool of questions posed by educators to evaluate the Ontoolcole ontology [56]. Table 1 summarizes the six search tasks that were proposed; note that there is a mix of clearly defined search tasks with more imprecise ones, since both types of questions were found to be significant in the previous study.

Concerning the procedure of the study, each participant carried out the aforementioned six tasks using one search system for the first three tasks and performing the remaining three with the other system. Similarly to TREC, a single sequence of search tasks was specified (TA-TB-TC-TD-TE-TF), while the starting task for each searcher was randomly determined. At the beginning of a task, a participant was provided with the task description and could then use the assigned system to find suitable tools. When she considered that the task was fulfilled, she had to fill a search questionnaire with all the obtained tools that she found appropriate. During this process, system logs kept track of the queries that were posed by participants. After completing the proposed

---

Note that a tool instance can correspond to various tool categories.
<table>
<thead>
<tr>
<th>ID</th>
<th>Task title</th>
<th>Brief description</th>
<th>Fuzziness</th>
<th># of tools</th>
</tr>
</thead>
<tbody>
<tr>
<td>TA</td>
<td>Meeting</td>
<td>A tool to support a meeting among co-located and distant students</td>
<td>Imprecise</td>
<td>5</td>
</tr>
<tr>
<td>TB</td>
<td>Whiteboard</td>
<td>A shared whiteboard to make annotations within group members</td>
<td>Precise</td>
<td>4</td>
</tr>
<tr>
<td>TC</td>
<td>Concept maps</td>
<td>A tool to author concept maps exportable to HTML format</td>
<td>Precise</td>
<td>4</td>
</tr>
<tr>
<td>TD</td>
<td>Share material</td>
<td>A tool to share the material employed in a course session with absentees</td>
<td>Imprecise</td>
<td>11</td>
</tr>
<tr>
<td>TE</td>
<td>Collaborative writing</td>
<td>An editor for writing documents in groups asynchronously</td>
<td>Precise</td>
<td>4</td>
</tr>
<tr>
<td>TF</td>
<td>Questionnaires</td>
<td>A questionnaire management tool for gathering students’ opinions</td>
<td>Precise</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 1
Overview of the search tasks.

Figure 6. Illustrative procedure of the study and collected data sources for a set of six participants (P1 to P6) performing a series of tasks (TA to TF).

search tasks, each participant had to respond to an exit questionnaire, prepared to get user feedback about the whole experience. Finally, a focus group session was held with a subset of the participants in order to gain further insight into their assessments of the experience. Fig. 6 graphically depicts the procedure of the study and identifies the employed data sources.

With this experimental design, it was possible to assess whether retrieval performance is better with Ontoolsearch or with Regain. Specifically, the F-
The harmonic mean of recall and precision, was taken as the response for evaluating the retrieval performance. The F-measure is a popular single-valued metric for interactive evaluations since the searcher’s goal is to select relevant results rather than ranking [34]. In order to calculate the F-measure, each task author was previously asked to inspect the tool dataset to obtain the tool instances that satisfy each search task; table 1 shows the number of instances that were considered suitable. The F-measure could then be computed for each observation, using the tools reported by the educators in the search questionnaires. Besides, the factors that were considered to influence the response were the system (the primary factor), the search task, the order of realization and the time employed to complete a search task. Since there are multiple groups to be compared, an ANalysis Of VAriance (ANOVA) model [20, pp. 250-253] was chosen in order to perform the quantitative analysis of the response.

Thus, an ANOVA model could be fitted to address the question of whether retrieval performance was significantly different using Ontoolsearch and Regain. In addition to this, it was desired to gain more insight about the study that could explain the quantitative results. With this aim, a mixed methods approach was followed to complete the analysis referred to different research categories. Specifically, a deeper comprehension of the search process both with Ontoolsearch and Regain was purposed in order to better understand retrieval performance measures. Second, it was desired to get feedback about the innovative user interface of Ontoolsearch. And third, identifying the strengths and weaknesses of Ontoolsearch was planned in order to improve this proposal in the future. Table 2 shows the correspondence of the aforementioned data sources with the evaluation goals of the study.

In this study, 18 educators participated; they are university teachers in Telecommunications (10), Computer Science (5), Philology (2) and Maths (1). All of

<table>
<thead>
<tr>
<th>Evaluation goal</th>
<th>Data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieval performance</td>
<td>Search questionnaire, System logs</td>
</tr>
<tr>
<td>Search mechanics</td>
<td>Exit questionnaire, System logs</td>
</tr>
<tr>
<td>User interface</td>
<td>Exit questionnaire, Focus group</td>
</tr>
<tr>
<td>Strengths and weaknesses</td>
<td>Exit questionnaire, Focus group</td>
</tr>
</tbody>
</table>

Table 2
Evaluation goals and data sources.
them have long experience in keyword searches, though not for searching tools in particular. Besides, most of them employ some kind of technological support in their learning settings. The study was carried out in a single session of two hours divided in two parts. The first hour was a tutorial so that educators could acquire a basic training of both Ontoolsearch and Regain. After the tutorial, the participants accomplished the proposed search tasks in the remaining hour. A few days after the experience, six volunteers participated in a focus group session that was moderated by an expert on qualitative research methods.

4.2 Retrieval performance results

After the completion of the study, the F-measure was calculated for each observation using the responses to search questionnaires. Then, an ANOVA model was fitted for the F-measure with the factors considered in the design. Since ANOVA models employ categorical predictors, the time employed factor was characterized as a categorical variable with five levels using the sample quintiles to fix the threshold of each level. In addition, two-factor interactions were also included in the model if they explained a significant part of the model variation.

Table 3 summarizes the data of the fitted ANOVA model from Matlab’s anovan, showing the sources of variability, the variation of the response due to each source and their p-values (last column). For a standard 95% confidence level, all the sources of variation were statistically significant except for the task and the order of realization. The ANOVA assumptions [20, pp. 334] were checked in order to assess the validity of the model. Specifically, the residuals passed the Kolmogorov-Smirnov [29] normality test and visual diagnostic tests [20, pp. 334-335] did not show trends in the mean or spread of the errors.

With the produced ANOVA model, a comparison of the mean F-measure of Ontoolsearch (0.7186) and Regain (0.5509) was performed. This test assessed that retrieval performance was better with Ontoolsearch, since the 95% confidence interval of the difference was (0.0811, 0.2542) with mean 0.1676. Indeed, this difference was highly significant since the measured p-value was 0.0002. This result shows that participants in the study performed better when using Ontoolsearch, thus evidencing that this system seems more appropriate for searching tools than Regain. To understand this improvement, more information can be derived from the model, analyzing the effect of the other factors.

Fig. 7(a) shows the influence of the task and the system in the mean F-measure. Retrieval performance was slightly better in tasks TB and TC when
Table 3
Summary of analysis of variance model for F-measure.

<table>
<thead>
<tr>
<th>Source</th>
<th>Degrees of freedom</th>
<th>Sum of squares</th>
<th>Mean square</th>
<th>F value</th>
<th>Prob &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>System</td>
<td>1</td>
<td>0.7227</td>
<td>0.7227</td>
<td>14.8412</td>
<td>0.0002</td>
</tr>
<tr>
<td>Task</td>
<td>5</td>
<td>0.5408</td>
<td>0.1082</td>
<td>2.2214</td>
<td>0.0597</td>
</tr>
<tr>
<td>Order</td>
<td>5</td>
<td>0.4763</td>
<td>0.0953</td>
<td>1.9562</td>
<td>0.0938</td>
</tr>
<tr>
<td>Time</td>
<td>4</td>
<td>1.1941</td>
<td>0.2985</td>
<td>6.1307</td>
<td>0.0002</td>
</tr>
<tr>
<td>System*Task</td>
<td>5</td>
<td>1.0780</td>
<td>0.2156</td>
<td>4.4277</td>
<td>0.0013</td>
</tr>
<tr>
<td>System*Time</td>
<td>4</td>
<td>0.8045</td>
<td>0.2011</td>
<td>4.1306</td>
<td>0.0042</td>
</tr>
<tr>
<td>Error</td>
<td>83</td>
<td>4.0415</td>
<td>0.0487</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>107</td>
<td>9.5730</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

using Regain. Conversely, higher improvements in the response were achieved with Ontoolsearch in the other tasks. In order to interpret these figures, submitted queries were extracted from system logs and subsequently analyzed. In the case of Regain, participants were able to find suitable keywords in tasks TB and TC. Indeed, asking for “whiteboard” retrieved all the relevant tool instances for task TB. Similarly, “concept map” and “mindmap” served to obtain all relevant responses to task TC. However, much more effort was required to provide appropriate keywords when facing an imprecise search task (TA and TD), so retrieval performance decreased in these cases. Noteworthy, the synonymity problem of keyword-based queries was specially relevant in task TE, since different terms and expressions were employed in relevant tools’ descriptions; for instance, collaborative editing, share documents or publish documents for public view were used by tool providers to describe asynchronous text editors (matching task TE).

With respect to Ontoolsearch, the influence of the task was less evident (see Fig. 7(a)). The analysis of submitted queries showed that participants were able to obtain appropriate results in most cases even with the imprecise tasks. This fact explains that retrieval performance was better when using this system. Nevertheless, some educators had some difficulties to compose their queries, negatively affecting the F-measure. In particular, new restrictions were sometimes incrementally added to previous queries so no results were obtained when submitted to the server. In other cases, participants wanted to express logical OR restrictions but they did not notice that AND semantics is always employed in Ontoolsearch.

The influence of the time employed in a search is shown in Fig. 7(b) for each system. With Regain, this factor had a limited impact in the response.
Figure 7. (a) Influence of the task and the system in the mean response. (b) Influence of the time employed in a search and the system in the mean response. Confidence intervals were calculated with a standard 95% level.

There is a trend to improve retrieval performance with time, although a slight decrease can be observed at the last level. The interpretation is that user effort to fulfil a search was higher when spending more time, thus obtaining more relevant results. Concerning Ontoolsearch, the influence of this factor was more significant. Shorter search times with Ontoolsearch led to better retrieval performance; query analysis showed in these cases that participants formulated one or two queries that matched most of the relevant tools for the proposed search tasks. Conversely, longer search times indicated educators had difficulties to fulfil the search task with Ontoolsearch, such as those described above.

The remaining factor to be analyzed is the order of realization. Although its influence in the response was not statistically significant, some trends were observed from the data. There is an improvement in the F-measure from the first task through the third. This could be explained by system learnability, allowing participants to increase their proficiency with the assigned system. At the fourth task participants switched to a new system and performance thus decreased. Better results were obtained at the fifth task, possibly due to system learnability as before. However, a performance decline was observed at the sixth task, attributed to educators' fatigue since they carried out this task at the end of the session.

4.3 Findings on search mechanics

The analysis of the remaining evaluation goals (see Table 2) is mostly based on qualitative data, interpreting collected feedback from educators. Beginning with search mechanics, Table 4 presents some findings derived from the exit
questionnaire along with some sample supporting comments\textsuperscript{2}. With respect to Regain, many participants found this system easy to use due to its similarity to other common search systems (see comment S1 in Table 4). Besides, they pointed out that query formulation is flexible and requires low effort (S2). However, educators had difficulties to find appropriate terms to fulfil the proposed search tasks, requiring to try with different sets of keywords (S3), as already detected in the query analysis. In addition, they were not confident of the results obtained with Regain, needing a careful revision to assess their suitability (S4).

When using Ontoolsearch, participants perceived that the process of query formulation was very different to that supported by Regain. Instead of providing a set of keywords, they had to compose a semantic query using the elements of the graphical user interface. Although educators were not accus-

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|}
\hline
System & Finding & Comment & Id \\
\hline
Regain & Easy to use & Previous training is almost not necessary since it is similar to web searchers like Google \cite{P2} & S1 \\
 & Flexible & I have more freedom to submit a question \cite{P5} & S2 \\
 & Problems with keywords & It is difficult to find appropriate keywords. It always seems there are fewer tools than with Ontoolsearch \cite{P14} & S3 \\
 & Requires revision of tool descriptions & It is necessary to carefully read a tool description in order to assess its suitability \cite{P16} & S4 \\
\hline
Ontoolsearch & Guidance facilitates the search & Guidance makes easier to find what I search \cite{P14} & S5 \\
 & Different paths for a search & There are multiple possible paths to perform a search. Very useful! \cite{P15} & S6 \\
 & Requires understanding the conceptual model & Sometimes it’s a bit difficult to begin a search if you are not sure what represents each node \cite{P1} & S7 \\
 & Comprehensible conceptual model & The best is the structuring in tasks, the relationships among tools and using graphs for searching \cite{P9} & S8 \\
\hline
\end{tabular}
\caption{Findings and sample comments on search mechanics. [EXIT QUESTIONNAIRE]}
\end{table}

\textsuperscript{2} In Tables 4, 5 and 7, [PX] refers to a comment expressed by participant X (from 1 to 18) in the study.
tombed to this way of searching, most of them rapidly got used to it. Indeed, some participants pointed out that it was more effective than using keywords (S5). In this sense, they perceived that Ontoolsearch guided the process of query formulation due to the underlying information structure. However, this guidance could have a negative effect if restricting users too much in their searches. This was not the case since educators were able to find multiple paths to conduct a search (S6). Perhaps the main challenge of Ontoolsearch was understanding the conceptual model employed to describe tools, specially at the beginning (S7). Despite this, the general feeling was that this model was comprehensible (S8), allowing participants to fulfil the proposed search tasks.

Interestingly, it was observed from the logs that educators required less time to compose a query with Regain, spending mean 0:45 seconds against 2:34 with Ontoolsearch. However, participants needed to formulate less queries with this latter system in order to fulfil a search task, submitting mean 3.5 queries per task against 11.0 with Regain. Moreover, educators were more confident with the results obtained with Ontoolsearch, since they rated their quality with a mean 5.3, while quality of results obtained with Regain was ranked with a mean 4.0 (using a six-level scale (1-least, 6-most) [Exit questionnaire]). Therefore, presented findings about search mechanics help to explain the observed improvement in retrieval performance with Ontoolsearch.

4.4 Findings on the user interface and other aspects of Ontoolsearch

Concerning the user interface of Ontoolsearch, Table 5 presents some findings derived from educators’ feedback. A key aspect was the use of graphs for query formulation; most participants rated very positively this approach (see for instance comment U1 in Table 5), although some found their usage a bit awkward, requiring a period of adaptation (U2). With respect to the results user interface of Ontoolsearch, it was highly appreciated by educators, since they were able to rapidly assess the suitability of obtained results (U3 and U4), in sharp contrast with Regain (see comment S4 in Table 4).

It was also purposed to get information about the learnability of Ontoolsearch as well as its ease of use in order to assess the appropriateness of the user interface design. Table 6 shows participants’ perception of these aspects for Regain and Ontoolsearch. Learnability of Regain was found higher since educators transferred their previous experience with other search systems. Nevertheless, ease of use of the two systems was comparable. Thus, the general feeling was that the user interface of Ontoolsearch was adequate for searching tools; for instance, comment U5 in Table 5 remarks this idea.
**Table 5**
Findings and sample comments on the user interface. [EXIT QUESTIONNAIRE] [FOCUS GROUP]

<table>
<thead>
<tr>
<th>Finding</th>
<th>Comment</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graphs OK for searching</td>
<td><em>I found the graphs lovely, spectacular, very well built because you can move the nodes in a way few tools let you do. Then, well, the graphs are very well set</em> [P5]</td>
<td>U1</td>
</tr>
<tr>
<td></td>
<td><em>At the beginning I didn’t like the graphs, but once I got familiar with the way they represent information they are quite adequate</em> [P2]</td>
<td>U2</td>
</tr>
<tr>
<td>Results UI appreciated</td>
<td><em>The best is the results window, showing all the possible ways to get a result. It’s very useful to understand a tool without reading its description</em> [P16]</td>
<td>U3</td>
</tr>
<tr>
<td></td>
<td><em>It gives you confidence that you get to good results: a set of tools that satisfies all specified requirements (the graph associated to each response is very good)</em> [P7]</td>
<td>U4</td>
</tr>
<tr>
<td>Adequate user interface</td>
<td><em>I was much surprised of the rapidness, easiness and intuitiveness of getting used to Ontoolsearch and performing searches</em> [P15]</td>
<td>U5</td>
</tr>
</tbody>
</table>

**Table 6**
Mean values of perceived learnability and ease of use (1-least, 6-most). [EXIT QUESTIONNAIRE]

<table>
<thead>
<tr>
<th>System</th>
<th>Learnability</th>
<th>Ease of use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regain</td>
<td>5.7</td>
<td>5.0</td>
</tr>
<tr>
<td>Ontoolsearch</td>
<td>4.8</td>
<td>4.8</td>
</tr>
</tbody>
</table>

Besides, participants in the study pointed out some limitations and other observations of Ontoolsearch that are shown in Table 7. One of these concerns was about the included conceptual model requiring users to understand it as well as to relate to their own conceptualizations (O1). Another limitation was the lack of help functionalities, since some participants had problems with Ontoolsearch. Previous query analysis detected some of these difficulties, supported with comment O2. In addition, formulation of keyword-based queries was proposed for Ontoolsearch, since it could be useful in some cases (O3). Further, this technique could be employed to browse the graph nodes (O4).

To conclude this analysis, participants remarked that search systems, such as those employed in this study, could be very useful to find suitable tools for their learning settings (O5), thus supporting the motivation of this proposal. Moreover, Ontoolsearch was considered very adequate for casual searching, helping to find tools for unexpected purposes (O6). Further, it was suggested to adapt Ontoolsearch to other non-CSCL scenarios that could be benefited...
Table 7
Findings and sample comments on other aspects of Ontoolsearch [EXIT QUESTIONNAIRE] [FOCUS GROUP] from this proposal (O7).

<table>
<thead>
<tr>
<th>Finding</th>
<th>Comment</th>
<th>Id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Needs understanding the conceptualization</td>
<td>The problem I see to all these systems that classify... is that your classification doesn't need to match with that included in the tool. In general, both classifications coincide, but there is always the exception, this 10% of cases in which they don't match [P18]</td>
<td>O1</td>
</tr>
<tr>
<td>Lack of help functionality</td>
<td>[I had a] problem with the interface in successive searches, because I forgot to erase previous searches [P3]</td>
<td>O2</td>
</tr>
<tr>
<td>Extension for keywords</td>
<td>Include a Regain-like module [in Ontoolsearch] to support keyword-based queries and obtain results or to take you to the most adequate node [P7]</td>
<td>O3</td>
</tr>
<tr>
<td></td>
<td>A textbox could be put here to insert the keyword, then you insert the keyword and this node is highlighted, and this node and this one: this means this keyword is in those three nodes [P18]</td>
<td>O4</td>
</tr>
<tr>
<td>Adequate for real practice</td>
<td>Until now, I hadn't considered searching tools for my classes. However, during the realization of the search tasks here I found some tools that could be used in my classes and, specially, in my research [P10]</td>
<td>O5</td>
</tr>
<tr>
<td></td>
<td>Regain can be useful if you’re certain about what to search. Ontoolsearch is very useful. It encourages you to explore and to discover other tools about other topics you didn’t know and you can like them and lead to use them in your courses. This seems very very useful to me [P7]</td>
<td>O6</td>
</tr>
<tr>
<td>Adaptation to other domains</td>
<td>I imagine they have considered this, but it is a possible future line for this work: this is very focused on CSCL settings, so it could be adapted to other learning environments [P1]</td>
<td>O7</td>
</tr>
</tbody>
</table>

5 Conclusions

Ongoing efforts on the CSCL domain have led to the delivery of a wide range of tools already available to support CSCL settings. Thus, educators require some means to find suitable tools for their intended CSCL scenarios. Unfortunately, current tool search facilities targeted to the learning domain commonly employ simple keyword matching techniques that are prone to obtain irrele-
vant responses. Further, educators are not conscious of existing tools and their educational applicability. Therefore, there is much room for improvement in tool searching and cataloguing.

This paper presented a new search system, named Ontoolsearch, intended for educators to discover tools for supporting CSCL settings. Ontoolsearch relies on the Ontoolcole ontology to annotate tools, enabling the processing of semantic queries referred to the elements defined in this ontology. Ontoolsearch has been designed to ease query formulation, offering an innovative direct manipulation interface to users. Indeed, the proposed user interface is the result of an iterative participatory design process involving both educators and developers. A prototype of Ontoolsearch has been already developed, making extensive reuse of available software libraries.

In order to evaluate this proposal, a formal comparison of the experimental system Ontoolsearch with the keyword-based search system Regain was carried out. 18 educators volunteered to perform six search tasks referred to real CSCL settings, using alternatively Ontoolsearch and Regain. Retrieval performance was analyzed fitting an ANOVA model, finding a highly significant improvement with Ontoolsearch, in spite of participants’ previous experience with keyword searches. Further, the mixed methods approach that was followed served to explain this result and to gain more insight about other aspects of the study. Thus, typical problems of keyword searches were detected (specially synonymity), while educators mastered the formulation of semantic searches with Ontoolsearch, being able to comprehend the conceptualization provided by the Ontoolcole ontology. Besides, participants rated the user interface of Ontoolsearch very positively, as well as its ease of use. Moreover, educators stated that Ontoolsearch is very useful to find tools for learning settings and to browse tools in a casual way.

Beyond these results, this work can be extended in a number of ways that should be tackled in the near future. Concerning the Ontoolsearch prototype, some limitations identified in the evaluation process could be addressed in a new release such as including help functionalities. In addition, further steps should be taken to spread this proposal to the CSCL community. For instance, the delivery of a specialized annotation tool for providers should facilitate the creation of Ontoolcole-based tool annotations. Moreover, new metadata extensions could be considered to allow searching for other features such as user ratings or even to adapt this proposal for other domains. Finally, the authors are currently working on the extension of this proposal to enable the integration of external tools previously found with Ontoolsearch in a tailorable collaborative learning system.
Acknowledgments

This research has been partially funded by the Spanish Ministry of Science and Innovation project TIN2008-03-23 and the Autonomous Government of Castilla and León, Spain, project VA106A08. The authors would also like to thank the multidisciplinary group GSIC/EMIC at the University of Valladolid and, very specially, all the educators who participated in the evaluation process.

References


[31] R. McGreal, Learning Objects: A Practical Definition, Instruction Technology and Distance Learning 1 (9).


