Abstract—Noctua is a tool to assist the Knowledge Acquisition and Collaborative Knowledge Construction processes. Noctua contains a virtual catalyst designed to facilitate the task of eliciting and validating knowledge. The virtual catalyst queries collaborators, proposing new knowledge, seeking confirmation to the knowledge already elicited, and showing conflicting opinions. Noctua takes into account collaborators’ profiles in order to ask them questions related to each one’s field of knowledge or interest. In this paper we present Noctua and the first experimentation we did using the tool.

Keywords - knowledge acquisition; virtual catalyst; collaborative knowledge construction

I. INTRODUCTION

In this paper, we present a web tool called Noctua (http://projetos.dia.tecpar.br/noctua/), which can be used for Knowledge Acquisition (KA) and Collaborative Knowledge Construction (CKC). Noctua has a virtual catalyst that aims at overcoming some obstacles inherent to the KA process such as lack of expert’s time availability and the difficulty in eliciting and representing knowledge. The catalyst also helps surpassing barriers innate to the CKC such as authorship registration and knowledge validation. The idea is to insert into the process a virtual member who plays the role of the catalyst, i.e. someone "whose talk, enthusiasm, or energy causes others to be more friendly, enthusiastic, or energetic" [3].

Noctua operates with conceptual knowledge, represented by Knowledge Pages [1] and procedural knowledge, represented by Production Rules [4].

The tool tracks users’ actions and log their collaborations. Additionally, the tool registers users’ opinions confirming or refuting something. This information is used by the virtual catalyst to choose which questions to ask and which ones not to ask to each user.

II. COLLABORATIVE KNOWLEDGE CONSTRUCTION

This section presents the fundamental concepts for the construction of the tool presented in section III.

A. The Knowledge and Knowledge Acquisition

For the purpose of this paper, knowledge may be considered as information combined with experience, context, interpretation and reflection, as defined in [5]. Milton presents in [1] a technical definition according to which “knowledge is the ability/skill/expertise to manipulate/transform/create data/information/ideas to perform skillfully/make decisions/solve problems”.

Knowledge Acquisition (KA) may be defined as “the transfer and transformation of problem-solving expertise from some knowledge source to a computer program” [6]. It is both an art and a bottleneck in the construction of knowledge-based systems [4].

The construction of a glossary of terms used by experts should be one of the first steps in the design of knowledge-based systems. According to Rolston [4], knowing experts’ vocabulary is a fundamental task in KA. Silva Junior et al. [7] state that “an adequate design of a cognitive system depends on the existence of a common vocabulary”.

One of the most popular KA techniques is the interview, despite criticisms on its efficiency [8]. It may be structured or unstructured, depending on its level of formalism and specificity.

Creating scenarios is another KA technique, in which experts are stimulated to explain their knowledge. Milton [1] suggests the creation of scenarios that depict or envision real situations. Scenarios with potentially inconsistent or missing information may stimulate experts to question them or to identify inconsistencies that were not recognized until then. This can be used by the knowledge engineer to make expert’s knowledge explicit.

B. Collaborative Knowledge Construction

Ramalho and Tsunoda [9] state that the Information and Communication Technologies created new spaces and forms for the construction of knowledge. Systems for collaboration (or collaborative software) make use of the Internet to foster communication and information organization, providing tools that facilitate the coordination inside groups of collaborators.
However, according to some researchers, the collaboration via web has some disadvantages. Pettenati and Ranieri [10] suggest that distance collaboration has deep social problems related to trust and reputation of the participants. It requires the development of a group culture and faces difficulties related to knowledge representation and management. The difficulty of representing the group and the competence of each member as well as the lack of face-to-face contact may weaken the sense of belonging and quickly lower the motivation to cooperate.

According to Nabeth et al. [11], the participation in communities of CKC is not, in fact, spontaneous, but driven by factors such as direct awards, gains from enhanced reputation or personal influence power, personal satisfaction with perception of their effectiveness and reciprocity. Participation requires a climate of trust, a sense of community, and a perception of recognition. These authors suggest a maximum requires a climate of trust, a sense of community, and a perception of recognition. These authors suggest a maximum effort to make visible the actions of each collaborator and their perceived value in the development of the collaborative process. This is what they call Social Translucence.

C. Collaborative Tool for CKC

According to Ackerman et al. [12], a system for CKC must have 3 basic characteristics: a tool for recording interviews, a discussion forum and a local memory. Noy et al. [13] presented a users’ reviews that compare the characteristics of several collaborative tools. They highlighted as desirable:

- An easy to use web interface;
- The capacity to show the reliability of each knowledge piece and each collaborator; and
- The capacity to allow disagreements and discussions about the knowledge under construction.

Lomas et al. [14] considered as important characteristics the possibility of synchronous and asynchronous collaboration as well as information about the authorship. Other important features cited by them are: adequate communication tools, easy-to-understand interface, image sharing, collaborative construction of documents and social interaction.

Noy and colleagues presented in [15], desirable characteristics of tools for collaborative development of ontologies. We think that these features would be also welcome at any CKC tool:

- The capacity of tracking the changes undergone by the knowledge during its construction and keeping all the related comments and discussions;
- The capacity to save old versions of knowledge, with the possibility of further changing it and to discard newer versions, as well as to compare two versions;
- Automatic identification of conflicting knowledge and mechanisms to resolve conflicts;
- Users with the power of mentoring, who have the final word on possible conflicts of knowledge.

D. Negotiation and Construction of Consensus

Collaborative construction of knowledge requires that all collaborators understand the shared knowledge representation and be able to express themselves by using it, to show their agreement or disagreement with other participants and to evolve, somehow, to a consensus or a final decision [2].

A collaborative tool must have rules about how knowledge might be proposed, changed or deleted. Dieng et al. [16] describe the CO4 protocol in which, when someone proposes a change in the knowledge, if there is no disagreement, the modification is performed. On the other hand, if there is any disagreement, the modification is not made. All participants are invited to comment and submit alternative proposals. The discussion ends when the rejected proposal is removed or when the disagreement is withdrawn.

According to Herrera and Fuller [17], negotiation is a key aspect in CKC because building consensus among the participants is a condition for the evolution of the knowledge within its life cycle. Those authors constructed a negotiation model that encompasses some predetermined actions such as: request for explanation, suggestions for modification, and adoption of a position by vote.

III. THE TOOL: NOCTUA

This section presents the general characteristics of our tool called Noctua, highlighting the main characteristics of the virtual catalyst.

A. The Noctua's Projects

Noctua allows every user to create KA/CKC projects. Every project starts out empty. This allows collaborators inside each project to work in their own area of knowledge and to express themselves in their own specific way, focus and desired depth.

B. Classification and Representation of Knowledge

Noctua allows the construction of two kinds of knowledge, according to a classification presented by Milton [1]:

- Conceptual Knowledge: tells what something is;
- Procedural Knowledge: tells how to do something.

Noctua represents Conceptual Knowledge using Knowledge Pages (KP), which describe the knowledge by means of natural language texts and pictures.

Procedural Knowledge is represented by Production Rules (PR) in the following format:

\[
\text{if } \langle \text{list of conditions} \rangle \text{ then } \langle \text{list of conclusions} \rangle
\]

Both representations were chosen because of their similarity to natural language which allows people unfamiliar to computers easily express their knowledge, as well as understand the information expressed in those formats.

C. Features of the Collaboration Tool

To overcome the difficulties presented in section II (subsection C), the Noctua allows the owner of each project to classify users as owners, tutors, and collaborators, each one with different permissions within a project.
Every collaborator may question the validity of a rule or entry. Questioned knowledge becomes invalid and must be discussed by the collaborators. If consensus is not reached, the matter may be finally resolved by a tutor.

Noctua allows all knowledge to be tagged. Collaborative tagging forms a folksonomy that reflects the collaborators' knowledge about the domain and can be helpful in building richer domain models in a consensual way [18].

Social Translucence is guaranteed by several aspects in the tool, such as: the registration and the disclosure of the authorship, chat between collaborators, and the disclosure of events such as creation and deletion of knowledge pieces.

Silva Junior et al. [7] establish a set of basic functions desirable in a groupware to support collaborative ethnography. If those functions are adapted to KA or to CKC, it is possible to see that Noctua embodies all of them:

- Creating, updating and closing KA/CKC projects;
- Recording users' profiles (and actively using them);
- Assigning users to activities (projects and roles);
- Recording notes and historical data (forums and instant messages);
- Creating documents (KPs and PRs);
- Supporting discussion and negotiation (by questioning knowledge and also in forums and instant messages);
- Supporting the awareness of the level of participation and contribution (by showing authorship statistics of each one's participation);

D. The Virtual Catalyst

The catalyst stimulates knowledge creation by asking collaborators questions and requiring their opinions (as shown further in this paper). Acting as a newcomer among experts, sometimes the catalyst asks impertinent questions, but sometimes its questions requires experts to rethink their concepts, so they may not only make their knowledge explicit but also wider.

The catalyst presents a screen containing a presumed knowledge (rule or some information within an entry). This knowledge, however, can be an exact copy of what is in the database or be a knowledge piece amended by the removal or the insertion of information related to other knowledge piece. In fact, when applied to rules, this technique creates scenarios and it can be used as described on section II-A.

Some of the possible forms of stimulation based on information in KPs are:

- Could you write something about <concept X>?
- What the text below is related to?
- In the entry on <concept X> it is written: <phrase A>. Do you agree with that?
- Which of the following statements applies to <concept X>?
- Is there a feature common to <concept X> and <concept Y>?

Concerning the rules, the virtual catalyst may act in several ways:

- Take a rule as it is (but not mentioning this to the user);
- Take two rules, mix some of their conditions and conclusions into a new tentative rule;
- Take a rule and delete some of its conditions;
- Take a rule and insert a condition or a conclusion from other rule.

The basic instigating question in these cases is: "See the rule below. Do you confirm it?"

Alternatively, a tentative set of conditions may also be used to ask the collaborator to think forward, not only confirming or refuting what is written, but trying to find a new valid conclusion. The question, in this case, is:

- Is there a possible conclusion if these conditions are fulfilled? (Then what?)

Asking for a new conclusion can also be done by using an established conclusion as a condition in a possible new rule. This leads to the creation of new layers of rules, i.e., rules whose conditions are conclusions from other rules.

IV. EXPERIMENTING WITH THE TOOL

Noctua was experimented by a group of ten people (faculty members from the Pontifical Catholic University of Paraná (PUCPR) and the Federal University of Technology - Paraná (UTFPR), and engineers from the Paraná Institute of Technology (TECPAR), all of them experts in computer programming. The experiment started with a face-to-face meeting in which the tool was presented to the participants. They defined the subject of the experiment: food and wine pairing. The idea was to describe wines and dishes in the Hyper Glossary and create rules for combining them properly.

The results are shown in Table I. The quantity of spontaneous knowledge eliciting (Spo) is compared to the quantity of those elicited by means of questions made by the virtual catalyst (Noc):

<table>
<thead>
<tr>
<th></th>
<th>Quantity</th>
<th>Noctua %</th>
<th>Spontaneous %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entries</td>
<td>12</td>
<td>17</td>
<td>83</td>
</tr>
<tr>
<td>Questions on entries</td>
<td>15</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Opinions on entries</td>
<td>12</td>
<td>17</td>
<td>83</td>
</tr>
<tr>
<td>Rules</td>
<td>38</td>
<td>16</td>
<td>84</td>
</tr>
<tr>
<td>Questions on rules</td>
<td>96</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Opinions on rules</td>
<td>48</td>
<td>29</td>
<td>71</td>
</tr>
</tbody>
</table>

It is possible to observe that the participants used the catalyst only a few times. They allowed Noctua to ask them 111 questions (an average of 11.1 questions per person) and yet about 1 in every 6 rules and entries was obtained after the
questions asked by the catalyst. This indicates that the catalyst is a potentially useful tool.

By analyzing the rules produced by the group, we observed that the process had a critical flaw: it was not carried out a preliminary step in which the input variables could be defined. Each participant created his own set of input variables instead of collaborating to create one only set. Some of them considered dishes as input variables and wines as conclusions, others made the opposite.

We also observed that 90% of the rules and 67% of the entries have one only author. This shows that each participant did not collaborate much to add information to what was started by another. They agreed, after the experiment, that this may have happened due to none of them being an expert in wines, so they did not feel comfortable to change each other's contributions.

A new experiment in which people will build knowledge about their own expertise is underway with a group of nursing students and teachers.

V. CONCLUSION

This paper presented a tool for knowledge acquisition and collaborative knowledge construction that uses KPs and PRs to represent knowledge. The tool uses the construction of glossaries, virtual interviews (by means of instant messages) and creation of scenarios as knowledge acquisition techniques. The proposed interface does not require users to have any computer programming skills. Terms inside an entry are constitued a hyper-glossary. Logical interconnections between entries and rules as well as timelines knowledge representation such as graphs showing the interconnections between entries and rules as well as timelines to show how the knowledge evolved along the collaboration process. Additionally, Noctua will be integrated with an inference engine so the rules can be tested on-line.

REFERENCES


