A Collaborative Approach to User Modeling within a Multi-functional Architecture

Pablo R. de Buen[1]*, Sunil Vadera[2], Eduardo F. Morales[3]

[1] Instituto de Investigaciones Eléctricas, Cuernavaca, México
[2] University of Salford, UK
[3] ITESM-Campus Morelos, México

Abstract. This paper describes a pragmatic collaborative user modeling approach based on three main components: a multi-functional knowledge base, an open user model, and a set of rules (i.e., the adapting criteria).

1 Introduction

One of the challenges of developing adaptive systems is to design architectures that can tailor the support given to the users, without restricting them from following their own preferred path. The cost of allowing greater flexibility is, however, (i) having less control and certainty about what the user is doing, or (ii) the need for augmenting the representation of the domain in order to take into account the different valid paths a user can follow. An alternative solution to this trade-off between control and flexibility is to involve the user in the process Self (1994). The user model, that is the main basis of the system decisions, can be built collaboratively with the user and can support the co-management of the dialogue (i.e., not requiring the system to make all the decisions). Additionally, the user model can be inspectable, promoting the reflection of the user upon its contents Self (1988) Bull and Pain (1995).

This paper presents a collaborative user modeling approach implemented in a multi-functional system called LacePro (Learning, Applying, and Consulting Established Procedures), which provides users with both alternatives: a directive system with adaptive characteristics, and a flexible system in which the user can define her own approach. This is within a collaborative environment in which the user provides the information the user model requires, has the possibility to inspect the user model at any time, and can reject or accept the system's suggestions.

2 LacePro

LacePro is a system in which professional engineers can learn, apply and consult established procedures. The architecture of LacePro enables a user to combine these three

* This work was developed under the support of CONACYT (Mexican government's council for science and technology).

tasks in order to satisfy her own requirements under a given situation. For instance, depending on how familiar she is with a procedure, a user can start applying the procedure to solve a problem, and then move to the consulting mode to inspect specific information, or move to the tutoring mode in order to have a full tutoring session about that procedure. When learning a procedure, she can select to receive a full tailored session that includes explanation of the steps and theoretical concepts, examples and evaluations; or she can choose to learn the procedure by solving problems or by navigating by her self through the domain knowledge; or by a combination of these two methods. To enable this flexibility, LacePros' architecture keeps the procedural structure of procedures. In LacePro, the domain knowledge is represented by a set of goal networks in which the nodes represent the steps of the procedure, and the links represent the flow of the procedure. Those theoretical concepts required to understand the basis of a step are stored in concept nodes that are linked to the step nodes.

In this representation, a step can be a procedure. The structure of the nodes contains all the information required to perform the tasks related with learning, applying and consulting the procedures. It also includes information about the user, which corresponds to an overlay user model (for more details about this representation consult De Buen et al. (1998)). The same single representation of procedures is used in LacePro for the tutoring, consulting and problem solving tasks. This makes it easy for the system (and user) to switch from one mode to another as desired. Additionally, in this way the user model can be shared by these three modules.

3 Adaptability in LacePro

Since the users of LacePro are professionals, the following conditions were considered for its design: (a) in general, people will use LacePro's tutoring capacity in order to learn a procedure that will enable her to solve a real problem (not just to learn a subject because it is part of a curriculum); and (b) it is expected that a person that uses LacePro (e.g., an engineer) will have some familiarity with the domain (at least she must have learned the general basis of the domain at school); this, in general, will ease the communication between the system and the user.

Based on the above conditions, a collaborative approach was selected. Two major considerations behind LacePro's collaborative approach are: (i) in principle, the system will consider as true all what the user says (e.g., that she knows a step or concept), and (ii) most actions proposed by the system (e.g., what to explain) can be accepted or rejected by the user.

In LacePro, the user model is comprised of three elements: (i) an overlay model over the network of the procedure, (ii) the characteristics of a user, and (iii) the structure of the procedure network, which can be traced to model a user's incorrect solution. These elements can be inspected by the user. This enables the user to know, at any moment, what the system has registered in relation to what she knows and/or has performed correctly. By accessing this information, the user may have a better understanding of the origin of the system's recommendations and, thus, have additional information for deciding whether to accept or reject them.

The user model is used to support the system in the following tasks: (a) to define whether or not to recommend an agenda for learning a procedure, before a user tries to solve a problem with that procedure; (b) to select the contents of this agenda; (c) to tailor the information that will be presented to the user when explaining the steps of a procedure and the theoretical concepts of the domain; and (d) to locate and remediate user errors.

4 Scenarios and Empirical Evaluation

LacePro's architecture was evaluated in two ways: (i) analysing the behaviour of the system under different scenarios (simulated data), and (ii) through an empirical study with twenty engineers. For these evaluations, a prototype of LacePro containing three real procedures for the wind design of structures was used. The scenarios (described in detail in De Buen (1998)) demonstrated the multi-functionality and adaptability features of the architecture. The empirical study showed that engineers liked this open and flexible approach.

5 Conclusions

The flexibility provided by both the multi-functionality and the collaborative user modeling approach of LacePro produce an environment, that, on the one hand, is able to satisfy the user requirements (by providing the applying, tutoring and consulting tools she requires) and preferences (e.g., enabling her to select how to combine these tools) and; on the other hand, can support her by proposing an agenda and the available information that best fits her state of knowledge and the characteristics of the procedure being used. In this collaborative approach, the system works as a proposer-assistant, and the user, who has access to the user model at any time, takes most of the decisions based on what the system recommends. Thus, LacePro's user modeling becomes an additional tool for the user, who has the flexibility to use it in the way she prefers.

References

Bull, S., and Pain, H. (1995). 'Did I say what I think I said, and do you agree with me?': Inspecting and questioning the student model. In *Proceedings of the World Conference on Artificial Intelligence in Education*, 501–508.

De Buen, P., Vadera, S., and Morales, E. (1998). A multi-functional knowledge based system to learn, apply, and consult procedures. *Expert Systems with Applications* 15(3/4):265–275.

De Buen, P. (1998). *An Adaptive Multi-Functional Framework To Learn, Apply and Consult Procedures, Doctoral dissertation.*

Self, J. (1988). Bypassing the intractable problem of student modelling. In *Proceedings of the Intelligent Tutoring System Conference (ITS-88)*, 18–24.

Self, J. (1994). The role of student models in learning environments. *IEICE Transactions on Information and Systems* E77-D(1):3–8.