AN ARCHITECTURE FOR KNOWLEDGE MANAGEMENT IN INTELLIGENT TUTORING SYSTEM

Simone Riccucci
Department of Computer Science, University of Bologna
Via Mura Anteo Zamboni 7, 40121, Bologna, Italy

Antonella Carbonaro
Department of Computer Science, University of Bologna
Via Mura Anteo Zamboni 7, 40121, Bologna, Italy

Giorgio Casadei
Department of Computer Science, University of Bologna
Via Mura Anteo Zamboni 7, 40121, Bologna, Italy

ABSTRACT
This paper discusses a general framework for knowledge acquisition and management in an intelligent tutoring system. This system is based on “Learning by performance errors” theory stating that in a given domain knowledge there is a set of constraints that must be satisfied in order to provide the correct solution to the problem. This paper addresses the issues of representing complex and generic information that applies to multiple domains. The proposed solution provides guidelines for both the system knowledge acquisition and management based on the natural language processing platform GATE (General Architecture for Text Engineering), inductive logic programming and constraint based paradigm.

KEYWORDS
Intelligent tutoring system, Constraint based tutor, Information Extraction System, Knowledge acquisition, Inductive Logic Programming

1. INTRODUCTION

ITSs are tools for assisted learning process. They leverage the knowledge of the domain and student models to implement adaptive tutoring that can approach the benefits of teacher to student interaction. The deployment of ITSs such as these ones (Anderson 1985, Koedinger 1997, Mitrovic 2001) have been demonstrating the efficacy of this technology. The scenario is composed by two actors, the teacher and the student, each of which interacts with a ITS module. The teacher provides the necessary knowledge to the ITS by means of an authoring tool and the student gets some questions from the ITS responding with an answer. ITS gives to a student the feedback on her solution showing the errors she made. This process is iterated until the student provides the right solution.

While ITSs have been proven useful in the learning process, they suffer from the disadvantage of being difficult to setup and tune for the following reasons:

• They require an expert in knowledge engineering in order to transfer the specific domain knowledge from teacher to computer;
• Currently ITSs are still complex to setup and manage because usually they require to implement ad hoc mechanisms depending on the domain they are made for.

The proposed architecture addresses the complexity of customizing the ITSs for specific knowledge domains by partitioning the knowledge codification and manipulation tasks in workable subtask some of
which can be aided by more advanced modules as the related research go on. In the following section the tutoring module and a part of knowledge acquisition module is described.

2. SYSTEM DESCRIPTION

The proposed architecture of tutoring and authoring module are illustrated in figures 1 and 3.

![Diagram](image)

Figure 1. Schema of tutoring module

We suppose a scenario where the student is asked to answer a question requiring a solution in a form of text. The system supports the student problem solving by providing feedback that relates to student errors. In order to accomplish this task the system needs for a knowledge base for extracting information from provided solution text and for correcting errors in such a solution.

The system knowledge base is made up of three components:

- a **domain ontology** describing the concepts and relations surrounding the domain concepts;
- a **set of parser rules** for identifying the entity in the provided solution;
- a **constraints base**, expressed in Prolog language, that a solution must satisfy in order to be correct.

The ontology and the parser rules are used by an Information Extraction System (IES) to tag the various parts of solution in order to recognize the entities corresponding to the concepts in the domain ontology. This process can be accomplished using ANNIE, a set of tools for Natural Language processing that is a part of the GATE framework (GATE 2005).

The component called “Tagged text to Prolog transducer” is used to obtain a version of the tagged solution in the form of Prolog ground facts. This set of facts must represent the information corresponding to the solution text and its structure must agree with the domain ontology (i.e. all the concepts name could be predicates name and relations between concepts could be represented as Prolog rules).

In this form the solution can be checked against the constraint base. Following the work of (Martin 2002 Suraweera 2004, Suraweera 2005) each rule in the constraint base has attached a feedback that explains the error concerning the constraint violation. The algorithm for error checking can be like that showed in figure 2. This task is accomplished by Solution Checker component.
The phase of knowledge acquisition can be divided in three major tasks: ontology building, creation of the set of parser rules and construction of constraint base. We will focus on automating the construction of constraint base even if there are some methods that can be used to automate the other two tasks. The overall scheme of knowledge acquisition system is illustrated in figure 3.

The problem of finding a general rule that must be satisfied in a domain, can be formulated as an Inductive Logic Programming (ILP) problem. In a general setting an ILP problem is formulated as follow: given a set of training positive and negative examples \( E \) and background knowledge \( B \), where \( B \) is a set of rules, find a hypothesis \( H \), expressed in some concept description language, such that \( H \) is complete and consistent with respect to the background knowledge \( B \) and the examples \( E \). In our case both the examples, background knowledge and hypothesis are expressed in Horn clause (Prolog language).

In our setting the positive, and optionally the negative, examples are given as a set of solutions to the same problem. Such solutions are then processed by the Prolog representation producer module in order to obtain a suitable representation to be processed in an ILP setting. The background knowledge is represented by the constraint set that initially is empty. This information are processed by another component called "ILP problem generator" that produces a set of data that can be directly processed by an ILP engine like ALEPH (ALEPH 2005). The ILP engine will produce a new set of rules representing the constraints related to provided examples. Such set must be validated before it can be added to the constraint database. The whole process can be repeated until all necessary constraints are found.

3. CONCLUSION

We have presented a framework for an Intelligent Tutoring System based on constraint paradigm and logic programming. This idea is inspired by the work of Mitrovic et al. used in various constraint-based ITSs (Mitrovic 2001). The main advantage respect to other framework is that each module can be developed as a separated field of research. Once we have a representation of solution for an ILP engine we do not need to
develop an ad hoc mechanism in order to find the constraints rule. The same reasoning can be applied for Information Extraction System. Furthermore the process of ontology building and parser rules creation should be automated as the research in this field go on. Some work for automatic acquisition of ontology have been done (Buitelaar 2003, Faure 1998, Maedche 2000, Navigli 2003) even if they need to be improved. Furthermore in GATE framework there is a plug-in for machine learning techniques that provides mechanisms to induce the parser rules starting from tagged text. Another advantage is the possibility of use an open answer approach for testing instead of a strongly structured answer as in (Suraweera 2004).

Currently there is almost nothing implemented of this framework and some issue can be considered only during the real implementation. However the clear separation of the stages involved in the various processes give us the strong theoretical basis of the research field involved like ILP and Information Extraction. We hope that this framework is general enough to address all type of domain knowledge even if initially it is more plausible to start with a domain of language programming that should result in less difficulties respect to domains that involve natural language.

REFERENCES

ACT-R a Cognitive theory about human cognition, http://act-r.psy.cmu.edu/

ALEPH: A Learning Engine for Proposing Hypotheses, http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph


GATE: General Architecture for Text Engineering, http://gate.ac.uk/


