Authoring Constraint-Based Tutoring Systems

Antonija MITROVIC[1], Stellan OHLSSON[2], Brent MARTIN[1]
and Pramuditha SURAWEERA[1]
[1]*Department of Computer Science and Software Engineering*
University of Canterbury, New Zealand
[2]*Department of Psychology, University of Illinois at Chicago*

This tutorial covers both the theory and practice of Constraint-Based Modeling (CBM). The first part of the tutorial introduces constraints and CBM as a theoretical foundation for Intelligent Tutoring Systems (ITSs). The second part covers ASPIRE, our new authoring system for developing constraint-based ITSs, and gives participants the opportunity to experience developing a small ITS in ASPIRE.

We will first introduce constraints as a way of representing domain knowledge. Currently, cognitive models typically cast declarative knowledge as consisting of propositions – knowledge units that encode assertions (which can be true or false) that support description, deduction and prediction. We have developed an alternative model of declarative knowledge that consists of constraints – units of knowledge that are more prescriptive than descriptive, and that primarily support evaluation and judgment. We present a formal representation of constraints and explain its conceptual rationale, and then introduce two applications of CBM. The first is the use of constraints as a basis for a machine learning algorithm that allows a heuristic search system to detect and correct its own errors. From this point of view, constraint-based learning is a form of adaptive search. This algorithm was originally developed as a hypothesis about how people learn from errors. We present the algorithm in some detail and briefly summarize applications to various problems in the psychology of cognitive skill acquisition.

Next we develop in detail the application of CBM to the design and implementation of ITSs. The constraint-based knowledge representation provides a novel way to represent the target subject matter knowledge, which has the advantage of directly supporting one of the main functions of expert knowledge in an ITS: to detect student errors. More importantly, the constraint-based representation provides a theoretically sound and practical solution to the intractable problem of student modeling. Finally, constraints and the associated learning algorithm provide detailed implications for how to formulate individual tutoring messages. We present multiple systems that follow this blueprint, together with empirical evaluation data.

In the second part of the tutorial we present our new authoring system (ASPIRE). The goal of ASPIRE is to make ITS authoring available to educators who have little technical knowledge of ITSs or programming in general. ASPIRE does this by providing extensive authoring support, such that the author, as far as possible, is always working at the domain knowledge level, not the programming level. We describe its architecture and functionality, as well as the authoring procedure it supports. The participants will then have hands-on opportunities to investigate ASPIRE closer, by using it to build a simple tutor.