The Effect of Problem Templates on Learning in Intelligent Tutoring Systems

Moffat MATHEWS[1] and Antonija MITROVIĆ[2]
Intelligent Computer Tutoring Group, University of Canterbury,
Christchurch, New Zealand
Email: {moffat[1], tanja2}@cosc.canterbury.ac.nz

Abstract: This paper proposes the notion of *problem templates* (PTs), a concept based on theories of memory and expertise. These mental constructs allow experts to quickly recognise problem states, almost instantaneously retrieve potentially vast amounts of domain-specific information, and seemingly effortlessly implement strategies to solve the problem. We investigate the role of problem templates in intelligent tutoring systems. An evaluation study was conducted in which PTs were created and implemented in SQL-Tutor. PTs were used to model students and make pedagogical decisions. Students using templates showed high levels of learning within short periods of time. Further evaluation studies are required to investigate the extent and detail of its effect on learning.

Introduction

The main goal of Intelligent Tutoring Systems (ITS) is to increase the effectiveness of learning. In this paper, we present the notion of Problem Templates (PTs), or high-level mental constructs used by experts to hold large amounts of relevant domain-specific information that is retrievable as a single chunk. Our goals were to examine the validity of such a construct, investigate if physical representations of PTs could be created, explore if pedagogical decisions within an ITS could be based on them, and if students' domain knowledge could be modelled using them.

Problem templates, as proposed in this paper, are chunks of domain-specific knowledge, compiled mentally by experts, used to solve commonly occurring problems in a particular domain. PTs are an extension to memory templates as proposed by the Template Theory (TT) [1]. Within PT slots, experts hold information to *recognise particular problem states, common strategies to solve* the problem (i.e. take the user from the current state to an intended solution state), and if required, a list of *tools and associated information* (or pointers to associated templates) required to implement these solutions. Because of PTs, experts have access to vast amounts of domain-specific information applicable to the current context. They are able to almost instantaneously recognise problem states within their domain, and seemingly effortlessly implement solutions. Comparisons of expertise traits [2] and various theories of memory [3] support the existence of PTs. In the driving instruction domain, students are taught templates (hill starts, 3-point turns etc) compiled by experts to increase their rate of learning. Experts in domains (e.g. chess [1]) compile mental libraries of templates. Templates are even used collaboratively in *game plays* to solve problem states between players in team sports such as hockey or football.

1. Introducing PTs to SQL-Tutor

We developed problem templates for SQL-Tutor [4], by categorising the 278 existing problems into 38 groups, whereby each group contained problems with the same solution structure. All relation and attribute names in solutions were then replaced by variables, leaving the generic SQL statements representing common solution strategies. We also developed a feedback message for each template, which is shown to the student on violation of the template. See Table 1 for two examples of SQL templates. As can be seen, template 1 is relevant for ten problems (whose ids are listed). The template also contains the general form of the goal SQL statement (third component), and finally the feedback message.

Table 1: Representation of SQL PT 1 and PT3

(1 (1 26 59 132 135 164 168 151 199 235) ("SELECT * FROM table" "Retrieve all attributes of one table"))
(3 (152 237 255 260) ("SELECT DISTINCT attribute(s) FROM table" "You want the details without duplicates (DISTINCT) of the attribute(s) of a table"))

When selecting a new problem, the control group students (using the original version of SQL-Tutor) were first presented with a choice of SQL clauses. To maintain the similar number of options for the experimental group, we generalised the 38 PTs into eight template groups, which were shown to students. SQL-Tutor highlighted one option as being preferred, but allowed the student to select a different one. In the latter case, the student was presented with a confirmation dialogue, alerting them to their deviation from the preferred choice. The student was then presented with the set of problems appropriate to their selection and their knowledge level. In this set, the preferred problem choice was highlighted; this choice could also be overridden by the user. After selecting the problem, the problem text with solution area was presented to the user. In both versions, the student could submit the solution at any stage. In addition to the feedback based on constraints, participants in the experimental group also received feedback associated with templates.

In the original version of SQL-Tutor, student models are represented as overlays on top of constraints; the experimental version used a similar process but with templates. Although constraint histories were recorded in student models for both groups for comparative analysis, they were used in pedagogical decisions only for the control version; template histories were used for the experimental group. Problem selection was based on the problem difficulty level (or template difficulty level), the student's knowledge scores, and the student's knowledge level.

2. Evaluation

An evaluation study was conducted at the University of Canterbury in May 2006 with volunteers from an undergraduate database course, who used SQL-Tutor either in the weekly lab sessions or at their leisure. Out of 68 students enrolled in the course, 48 logged into SQL-Tutor and completed the pre-test. Participants were randomly allocated into two groups: the control (23) and the experimental (25). Pre-tests results showed no significant differences between the two groups: 42% (6%) and 50% (7%) for the experimental and control group respectively.

Unfortunately, only ten students completed the post-test, and consequently the data is insufficient for in depth comparisons. However, the experimental group students needed significantly shorter time (58% of the time needed by the control group students) to solve the same number of problems (p=.055), with fewer attempts. Learning curves for the two groups are very similar, and closely fit the power curves: $y=0.0995x^{-0.384}$ for the control group ($R^2=0.92$), and $y=0.1044x^{-0.4079}$ for the experimental group ($R^2=0.89$). This indicates that both groups learned domain-specific knowledge at similarly high rates, and did so equally well.

Some restrictions to this research exist. First, the study was relatively short (four weeks). Generally, under non-ITS instruction (e.g. sports), students take much longer periods of time to compile (create and organise) PTs. Compiling templates however, takes them from being intermediate users to mastery. Second, PTs taught by human tutors are generally compiled by combining the knowledge of many experts and refined over a period of years, while PTs used in this study were compiled by a single person. Third, although PTs can be learned implicitly and automatically, they are generally taught explicitly. In this study, students did not undergo formal teaching of PTs. In spite of these restrictions, the experimental group learned domain-specific knowledge at an equally high rate, and attempted and solved the same number of problems in shorter periods than their counterparts in the control group.

The goals of this research were to examine the validity of PTs, explore if physical representations of PTs could be created and implemented in an ITS, investigate the use of PTs in student modelling and pedagogical decisions, and examine its effects on learning. Relevant prior research and real-life examples support the notion of PTs. Physical representations of problem templates were created in a complex, rich, and relatively open-ended domain (SQL). Problem templates were used for student modelling, problem selection and feedback generation. Lastly, an evaluation study was conducted in which participants used either a control version of SQL-Tutor (using only constraints), or the experimental version (using templates). Preliminary results were also promising, showing equally high rates of learning in both groups.

Future work includes conducting an evaluation study over a longer period of time to collate sufficient data, exploring applicability of PTSs to other instructional domains, and conducting scientific studies to comprehend the processes used by human tutors and students when using PTs.

References

[1] F. Gobet and H. A. Simon, Templates in Chess Memory: A Mechanism for Recalling Several Boards, Cognitive Psychology, 31, 1-40, 1996.

[2] J. Shanteau, The Psychology of Experts: An Alternative View, in Expertise and Decision Support, G. Wright and F. Bolger, Eds. NY: Plenum Press, 11-23, 1992.

[3] F. Gobet, Expert Memory: A Comparison of Four Theories, Cognition, 66, 115-152, 1998.

[4] A. Mitrović, An Intelligent SQL Tutor on the Web, Artificial Intelligence in Education, 13, 173-197, 2003.