Ogata, H. et al. (Eds.) (2015). Proceedings of the 23rd International Conference on Computers in Education. China: Asia-Pacific Society for Computers in Education

Developing an Embedded Tutor for On-the-job Training

Jill DE JONG*, Antonija MITROVIC & Moffat MATHEWS
Intelligent Computer Tutoring Group, University of Canterbury, New Zealand
*jill.dejong@pg.canterbury.ac.nz

Abstract: We describe Chreos Tutor, a constraint-based tutor embedded into Chreos, an existing business software system. The goal of the tutor is to teach users new to Chreos how to complete realistic tasks. The student interface is the combination of a new tutoring screen and existing Chreos data input screens. We have conducted a pilot study investigating the usability of Chreos Tutor, the results of which show that the tutor is promising.

Keywords: embedded tutor, on-the-job training, pilot study

1. Introduction

Traditional on-the-job training for software typically involves human tutoring, training videos, documentation and open-ended exploration of the software. All of these tools have their limitations. Human tutoring is expensive, while training videos and documentation lack guidance and feedback. The risk with open-ended exploration is that some essential skills may be missed. The main focus of users in a job situation is on successful completion of their tasks in minimal time (Rieman, 1996).

Intelligent Tutoring Systems (ITSs) provide an alternative to conventional on-the-job training. Embedding the ITS in the software allows users to practice realistic tasks in the actual environment, at their convenience. A number of ITSs have been embedded in a range of existing software, such as MACSYMA Advisor (Genesereth, 1979), Geometer Sketchpad and Microsoft Excel (Ritter & Koedinger, 1996), a military information system (Cheikes et al., 1998) and Microsoft Access (Risco & Reye, 2012). This paper describes the development of Chreos Tutor, a constraint-based tutor embedded into the Chreos business system, to assist users to learn realistic tasks. Chreos (http://www.chreos.com) is a business system developed by Wild Software Ltd, with several integrated modules, the most commonly used of which are Inventory and Debtors. The Inventory module provides functionality to manage products and services. The Debtors module enables a business to enter and keep track of data relating to its clients. Chreos implements double entry bookkeeping principles for all financial transactions. New Chreos users are often trained by one or more support personnel. Remote training is also available, where support staff can see the user's desktop and talk them through the process via telephone. There are also some training videos and a range of web-based help documentation.

Chreos Tutor is developed in the ASPIRE authoring system (Mitrovic et al., 2009). In the next section, we describe the development process and the user interface, followed by a description of the pilot study we have recently performed.

2. Modeling Chreos Tasks in ASPIRE

ASPIRE is a system for authoring and deploying constraint-based tutors. ASPIRE-Author provides a web-based interface that facilitates the modeling of instructional domains, the structuring of tasks and their solutions, student interface design, the entry of tasks and solutions, constraint generation and tutor deployment. ASPIRE-Tutor is a web-based server that makes the deployed tutor available to students.

Chreos Tutor teaches two tasks from the Clients module. The first task involves making financial adjustments to debtor balances, which is done via the Debtor journal screen. The second task is to enter client orders. We modelled each task as a separate non-procedural domain in ASPIRE, and developed the domain ontologies. The central concept in both domain ontologies is *Input Field*, an

abstract concept containing a *Value* property to represent the value a user selects or enters into that input field. Each simple screen component that needs to be filled out is modeled as a sub-concept of *Input Field*. The debtor journal screen components can all be modeled in this way. The Client Orders domain ontology contains additional concepts to cater for a grid. The Client Orders screen requires only a single *InvoiceGrid* sub-concept to represent the list of items included in the order. Each item forms a row in the grid, so the *InvoiceGrid* has a *Rows* property which is related to the *Row* concept. The properties of the *Row* concept represent the fields critical in determining if the item details are correct. *ITEMREF* is a system generated identifier unique to an item, *QUANTITY* is the quantity ordered by the client and *UNITAMTINC* is the unit selling price including Goods and Services Tax.

The solution structure specifies the list of solution components. Each of these components has a label, the corresponding concept(s) from the domain ontology and an element count to indicate the number of elements it can contain.

ASPIRE automatically generates an HTML interface from the domain, and the problem/solution structures. The author can choose to use this default interface, or upload one or more applets. Chreos Tutor provides a student interface inside Chreos so does not use either of these interface options. The student interface communicates with ASPIRE via XML remote procedure calls.

We have defined 12 tasks for Debtor Journal, and 16 for the Client Orders domain. All task descriptions refer to and are consistent with the tutoring database, which is an amended version of a database used for Chreos sales demonstrations. In a normal Chreos entity, the database is updated when input is processed. This cannot happen with the tutoring database, because the ITS tasks are fixed and can be completed by multiple users at different points in time. The database needs to remain in the state that existed when the tasks and their solutions were originally determined. All Chreos input screens have a *Save* button and normally when this is clicked the database is updated for the new input. The Save button on screens relevant to Chreos Tutor tasks are hijacked by the tutor to instead send the task solution to ASPIRE and return feedback to the student interface.

Dates are important pieces of data in Chreos. This can be the date or month a transaction belongs to, as well as the date the transaction is entered. The first type of date is usually selected by the user, whereas the second type comes from the computer date. Some modules in Chreos have their own system date, which allows those modules to be in a different month to other modules and in an earlier month than the real world. This allows users to enter transactions into the appropriate month in Chreos even if that month is finished in the real world. The Client module has its own 'current system date'. Transaction dates in Chreos screens usually have a default value. For Chreos Tutor purposes the Debtor Journal transaction date defaults to the 'current system date' for the Client module. The Client Orders screen contains 3 date fields. For Chreos Tutor purposes, the Client Orders screen opens with the transaction date set to the computer date, the *'Deliver by'* date set to the computer date plus 1 day and the *'Not Before'* date equal to the transaction date.

ASPIRE generates constraints based on the information provided by the author in terms of the domain ontology, task/solution structures as well as the list of tasks and their solutions. For each mandatory solution component, ASPIRE generates constraints that check whether the student has entered a syntactically correct value and whether that value matches the value in the task solution. For each optional solution component, generated constraints check that a value is present in the student solution only if it exists in the task solution, that such a value is syntactically correct and matches the value in the task solution. The Debtor Journal domain contains 20 constraints. Additional constraints are generated for the *InvoiceGrid* and *Rows* concepts in the Client Orders domain. These ensure that the student solution contains the correct number of rows, that each row in the student solution matches a row in the task solution and that each row in the task solution matches a row in the student solution. Together these additional constraints check that the solutions are an exact match, without limiting the order in which items are entered. The Client Orders domain contains 35 constraints.

The student interface is embedded into Chreos. The major part of the interface is the Chreos Tutor screen, which controls all interaction between the user and the CBT in both directions. When this screen opens, it displays some general information about the tutor. Once a user clicks the *Connect* button to connect to the tutor, the general information screen is replaced by the tasks screen. From there users can select the type of task they want to learn, access task-type help documentation and open up an additional screen giving them step by step instructions for using the tutor. The next step is to select a task from that domain and open the appropriate Chreos input screen in order to perform the task. Users receive feedback indicating whether the screen they have opened is correct for the selected task.

The user completes the task using information provided in the task description, and can submit the solution by clicking the *Save* button. This button has been hijacked so that it behaves like the *Submit* button in typical ASPIRE tutors. The feedback level automatically increases each time the *Save* button is clicked. There are four graduated levels of feedback, with the fourth level being the full solution. Feedback starts out at the *Quick Check* level, which indicates whether the solution is correct, or specifies the number of input fields with errors. The *Hint* level provides a hint about how to correct the error in one particular input field. The third level of feedback is *Show all errors*, which provides hints about each error. All incorrect input fields are highlighted with a red border and when a user hovers their mouse over one of those fields, the hint also shows as a tooltip.

3. Pilot Study and Future Work

We have recently conducted a pilot study investigating the usability of Chreos Tutor, with five members of the Intelligent Computer Tutoring Research Group, two Chreos users and one Accounting lecturer. They were asked to attempt three particular tasks of each type, and complete a questionnaire. The questionnaires asked participants to rate certain aspects of the tutor and also provide open-ended feedback on those aspects. The amount of time participants spent using Chreos Tutor ranged from 34 to 62 minutes, with an average time of 51 minutes. Six participants successfully completed either 5 or 6 of the set tasks, one completed 3 tasks and one did not submit a solution to any of the set tasks.

One problem with embedding a tutor into existing software is that the tutor must use the same type of interface, rather than an interface designed specifically for instruction purposes. Analysis of the questionnaires indicates that the participants who were not familiar with accounting and Chreos struggled with some interface features Chreos provides. Novice users must learn how the screen components work in addition to understanding which data belongs in which input fields. Some participants did not realise that dates only needed to be altered if the default value was inappropriate for the task. Some did not understand how to enter the client. There were also problems adding items to the order. Some participants indicated that instructions would benefit from being less wordy and more visual. There were also some issues with task descriptions, particularly with respect to dates and other default settings. The timing and style of feedback was generally found to be very helpful. Participants particularly liked the highlighting of incorrect fields and seemed to favour the *Show all errors* or the *View full solution* levels of feedback.

We have made some small changes to improve the usability of Chreos Tutor. The most important is an introductory video for each domain, showing how many of the input fields should be filled out. The appropriate video is available to users once they have selected a task type. The formatting of instructions and feedback has been improved and images are now incorporated where appropriate. The wording of task descriptions is in the process of being amended to increase user understanding.

The size of the Chreos user base means that there will not be enough real novice Chreos users for a full evaluation study of Chreos Tutor. Therefore we plan to evaluate the system in August 2015 with students enrolled in a course on accounting information systems.

References

Cheikes, B. A., Geier, M., Hyland, R., Linton, F., Rodi, L., & Schaefer, H. P (1998). Embedded Training for Complex Information Systems. *Proc. Intelligent Tutoring Systems*, Lecture Notes in Computer Science, pp. 36-45.

Genesereth, M. R. (1979). The Role of Plans in Automated Consultation. *Proc. 6th Int. Joint Conference on Artificial Intelligence*, Vol. 1 (pp. 311-319).

Mitrovic, A., Martin, B. Suraweera, P., Zakharov, K., Milik, N., Holland, J., McGuigan, N. (2009). ASPIRE: an authoring system and deployment environment for constraint-based tutor. *Artificial Intelligence in Education 19(2)*, 155-188.

Rieman, J. (1996). A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction, 3(3)*, 189-218.

Risco, S., & Reye, J. (2012). Evaluation of an intelligent tutoring system used for teaching RAD in a database environment. *ACE 2012: Proc. 14th Australasian Computing Education Conference*, (pp. 131-140).

Ritter, S., & Koedinger, K. (1996). An architecture for plug-in tutor agents. *Journal of Artificial Intelligence in Education 7(3-4)*, (pp. 315-347).