Chen, W. et al. (Eds.) (2016). Proceedings of the 24th International Conference on Computers in Education.
India: Asia-Pacific Society for Computers in Education

An Embedded Constraint-based Tutor for On-the-Job Training

Jill DE JONG[a], Antonija MITROVIC[a*] & Moffat MATHEWS[a]
[a]*Intelligent Computer Tutoring Group, University of Canterbury, New Zealand*
*tanja.mitrovic@canterbury.ac.nz

Abstract: We present Chreos Tutor, a constraint-based tutor embedded into Chreos, an existing business software system. The goal of Chreos Tutor is to teach users new to Chreos how to complete realistic tasks. In Chreos Tutor, the student interface is the combination of a new tutoring screen and existing Chreos data input screens. We conducted a study investigating the effect of feedback Chreos Tutor provides, which shows that the feedback resulted in a significantly higher learning gain. An experienced Chreos user found the tutor to be a preferred training option for new users.

Keywords: embedded intelligent tutoring system, on-the-job training, evaluation study

1. Introduction

Traditional on-the-job training for software typically involves human tutoring, training videos, manuals and open-ended exploration of the software. All of these tools have their limitations. Human tutoring can be tailored to each individual user, but is expensive and can be inflexible in terms of timing. Training videos and documentation are normally less expensive options that can be used when convenient, but are less effective because of lack of guidance and feedback. The risk with open-ended exploration is that some essential skills may be missed. A field study investigating the learning behaviour and attitudes of computer users in everyday working situations (Rieman, 1996) indicated that the main focus of software users in a job situation is on successful completion of their tasks within the minimum time frame. Time pressure tends to favour a ‘just in time’ task-driven approach to acquiring software skills.

Intelligent Tutoring Systems (ITSs) provide an alternative to conventional on-the-job training tools for software users. Embedding the ITS in the software allows users to practice on realistic tasks in the actual environment, at their convenience. ITSs tailor tasks for incremental learning of software features or task types, and provide adaptive feedback.

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 and Koedinger, 1996), a military information system (Cheikes et al., 1998) and Microsoft Access (Risco and Reye, 2012). DM-Tutor (Amalathas, Mitrovic and Ravan, 2012) is a constraint-based tutor developed using the ASPIRE authoring system (Mitrovic et al., 2009). This tutor was embedded within an existing Management Information System (MIS) for oil palm plantation management, with the aim of improving decision making skills by providing scenario-based training using actual data and plantation conditions.

We present Chreos Tutor, a constraint-based tutor embedded into an existing business software system to assist users to successfully complete realistic tasks. Similar to DM-Tutor, Chreos Tutor is also developed in ASPIRE. Its student interface is the combination of a new tutoring screen and an existing data input screen inside the Chreos Business Software application (referred to as Chreos). Chreos Tutor operates in a similar manner to Motherboard Assembly Tutor (MAT) Westerfield, Mitrovic and Billinghurst, 2013), with Chreos obtaining problems and feedback from the tutor and providing information on user actions to the tutor via remote procedure calls. However, the MAT teaches how to assemble components on a computer motherboard via an Augmented Reality interface rather than an existing workplace system.

This paper is organized as follows. In the next section we introduce Chreos Business Software, which provides the context for workplace learning. Section 3 discusses the development of Chreos Tutor in ASPIRE. We have recently performed a study, in which we focused on the effect of feedback provided by the tutor. The study is presented in Section 5, followed by the results and conclusions.

2. Chreos Business Software

Chreos[1] is a modular, integrated business system. Its modules include Point of Sale, Debtors or Clients, Creditors, Inventory and General Ledger. These modules are integrated where relevant in order to minimize duplication of data input and maintain data consistency. The most commonly used modules are Inventory and Debtors. The Inventory module provides a business with the functionality to manage the products and services it is buying and selling. The Debtors module enables keeping track of client data. The processes for entering transactions and viewing data are reasonably consistent throughout Chreos, so that knowledge acquired in one module is applicable in other modules.

Chreos implements double entry bookkeeping principles for all financial transactions. This can be implemented explicitly, where the user enters both sides of the transaction in a screen. However, with most screens users enter the minimal amount of data. Chreos then carries out any necessary calculations and generates all the appropriate bookkeeping entries. New Chreos Service Packs are released on a regular basis, usually at a rate of 1 to 2 per year. The software is written in Delphi and runs on a MS Windows operating system. It uses Client-Server architecture and the Firebird relational database for data storage.

Wild Software Ltd, the developers of Chreos, provide training and support to all users, which often incorporates training sessions run 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. Each client of Chreos can also set up a Practice Company to use for training, which is either a copy of their current database or of an earlier backup. This enables users to try out various tasks, without impacting live data. Beyond that there are some training videos and a range of web-based help documentation.

The background and attitudes of new users of software can affect the learning experience. The size of the Chreos user base limits the chances of a new user having previous Chreos experience. A new user can be an existing employee of a business that decides to install Chreos, or a new employee of a business already using Chreos. In the first case, there are usually multiple people having to learn about Chreos, with at least some of them perceiving that it has advantages over the previous system. In the second case, there is a single new user who may have preconceived ideas of how Chreos should behave from previous systems. An ITS that allows users to practice real world tasks in the actual Chreos environment at their convenience would be a valuable training tool in both situations.

3. Modelling Chreos Tasks in ASPIRE

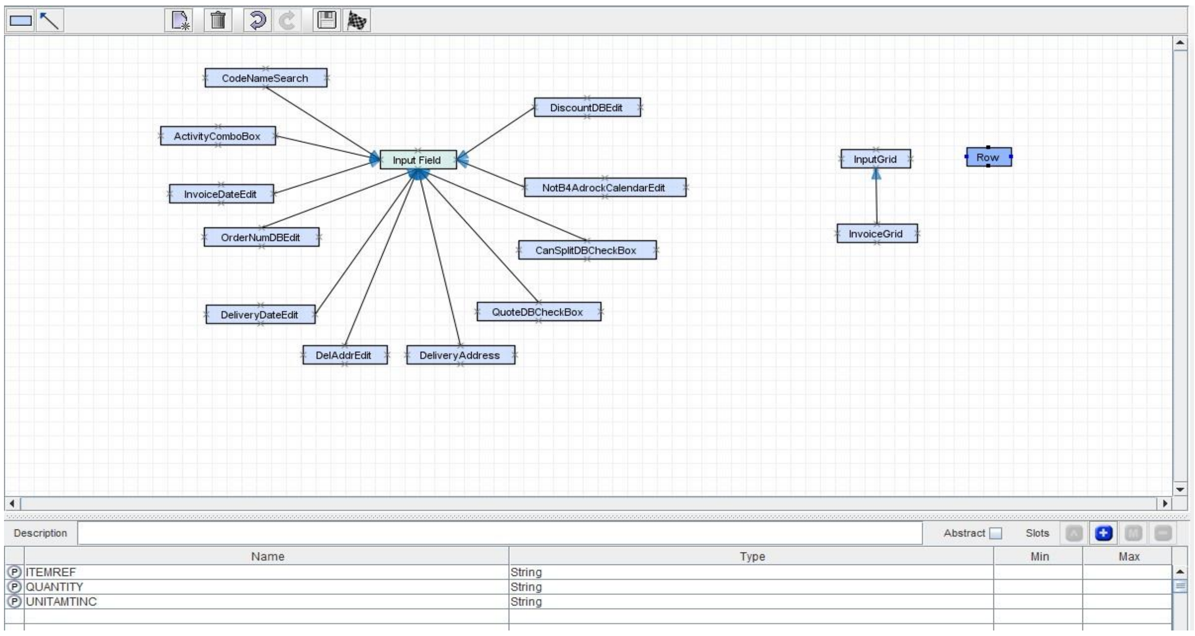
ASPIRE is a system for authoring and deploying constraint-based tutors. ASPIRE-Author provides a web-based interface that supports authors in developing new tutors, by specifying domain ontologies, problem/solution structure and student interfaces as well as entering problems and their solutions. ASPIRE automatically generates constraints which the author can modify, if necessary. ASPIRE-Tutor is a web-based server that deploys tutors to students.

Chreos already breaks business processes down into tasks, so for ease of learning Chreos Tutor adopts the same definition of tasks. A task only involves a single Chreos screen. We selected 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. When users enter data into Chreos, the order in which fields are entered is not critical. The critical thing is that when the user clicks the Save button, everything they have entered into the screen is correct. Consequently each of these tasks is modeled as a separate non-procedural domain in ASPIRE.

[1] http://www.chreos.com

Figure 1 shows the Client Orders domain ontology. The central concept in the domain ontology 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 correctly filled out is modeled as a sub-concept of Input Field. Please note that the arrows show the specialization relationships visually; for example, ActivityComboBox is a sub-concept of Input Field. Each sub-concept inherits the properties of its parent, so no additional properties are necessary. 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. It is possible to have multiple grids in a screen, so the InputGrid concept represents a generic grid component in an input screen. 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 are shown in Figure 1 (as that concept is the selected one). These properties 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.



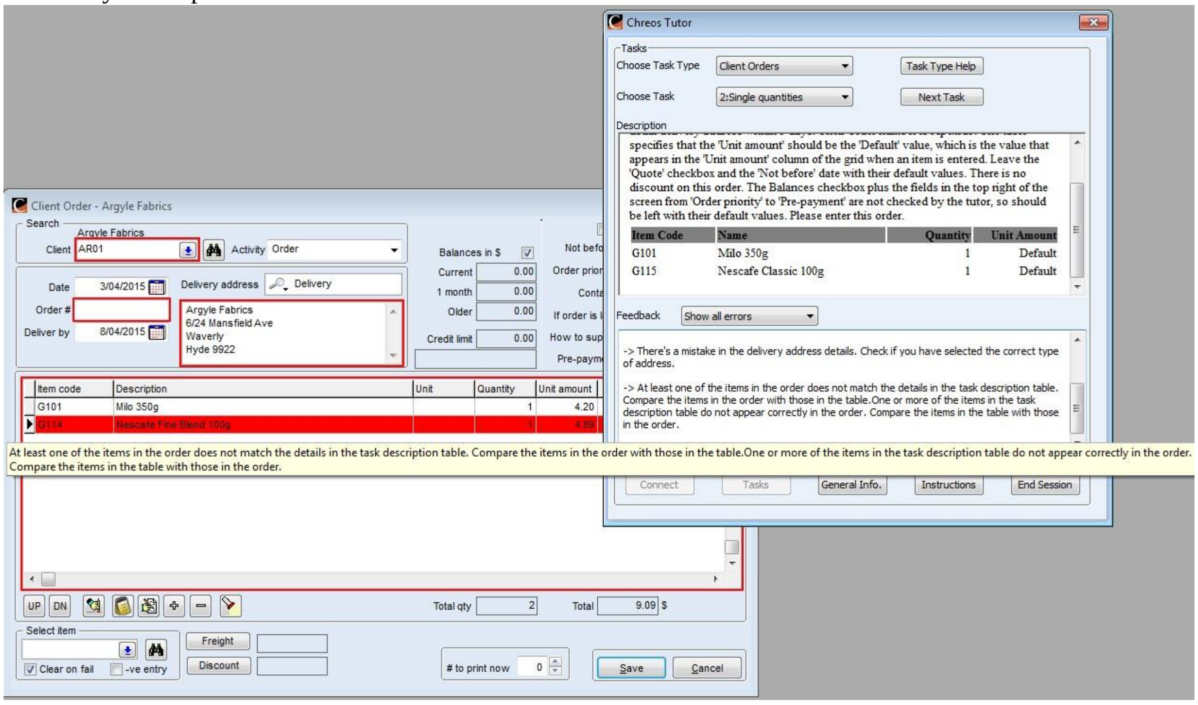
Name	Type	Min	Max
ITEMREF	String		
QUANTITY	String		
UNITAMTINC	String		

Figure 1. Client Orders Domain Ontology

The solution structure for each domain 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. We have defined 12 tasks for the Debtor Journal domain, 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.

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 22 constraints, while 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 tutor. When this screen opens, it displays general information about the tutor. Once a user clicks the Connect button, the tasks screen is shown, allowing users to 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.



Choose Task Type: Client Orders

Choose Task: 2:Single quantities

Description

specifies that the 'Unit amount' should be the 'Default' value, which is the value that appears in the 'Unit amount' column of the grid when an item is entered. Leave the 'Quote' checkbox and the 'Not before' date with their default values. There is no discount on this order. The Balances checkbox plus the fields in the top right of the screen from 'Order priority' to 'Pre-payment' are not checked by the tutor, so should be left with their default values. Please enter this order.

Item Code	Name	Quantity	Unit Amount
G101	Milo 350g	1	Default
G115	Nescafe Classic 100g	1	Default

Feedback: Show all errors

-> There's a mistake in the delivery address details. Check if you have selected the correct type of address.

-> At least one of the items in the order does not match the details in the task description table. Compare the items in the order with those in the table.One or more of the items in the task description table do not appear correctly in the order. Compare the items in the table with those in the order.

Argyle Fabrics
Client: AR01
Activity: Order
Date: 3/04/2015
Order #:
Deliver by: 8/04/2015
Delivery address: Delivery
Argyle Fabrics
6/24 Mansfield Ave
Waverly
Hyde 9922

Balances in $ ✓
Current: 0.00
1 month: 0.00
Older: 0.00
Credit limit: 0.00

Item code	Description	Unit	Quantity	Unit amount
G101	Milo 350g		1	4.20
G114	Nescafe Fine Blend 100g		1	4.89

At least one of the items in the order does not match the details in the task description table. Compare the items in the order with those in the table.One or more of the items in the task description table do not appear correctly in the order. Compare the items in the table with those in the order.

Total qty: 2
Total: 9.09 $
✓ Clear on fail
-ve entry
to print now: 0

Figure 2. Client Orders Task 2 with *'Show all errors'* Feedback

Students complete tasks in the input screen, and submit solutions by clicking the Save button (Figure 2). The level of feedback automatically increases each time the Save button is clicked. There are four graduated levels of feedback, with the fourth level being a screenshot of the ideal 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. Those fields are highlighted with a red border in the input screen. The Hint feedback level provides a hint about how to correct the error in one particular input field. This field is highlighted with a red border in the input screen. The third level of feedback is Show all errors, which provides hints about how to correct each input field that contains an 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. Feedback remains at this level until a new problem is selected or the user selects a different option.

4. Study

Ideal evaluation of Chreos Tutor would include two groups of users, one learning about Chreos using the existing training procedure (videos, manuals and sessions with support personnel), and the other group using Chreos Tutor. However, this kind of evaluation is not possible, due to the small number of new Chreos users. Instead, we conducted a study involving volunteers recruited from the University of Canterbury course on Accounting Information Systems. The students had previous accounting knowledge and used a different accounting package in their course. Therefore, they approximate the type of person who would be a novice user of Chreos.

The study was based on an experimental design with repeated measures involving two conditions, consisting of a pre-test, a problem-solving phase and a post-test. The participants were randomly allocated to one of the two versions of Chreos Tutor: the experimental group interacted with the full tutor, while the control group had no feedback. Upon submitting solutions, the control group participants could view the ideal solution only. This kind of control condition resembles a classroom situation, when students typically do not receive individual feedback on their solutions, but are eventually able to see the ideal (i.e. teacher's) solution. In such a way, the control group participants could compare their solution to the correct one.

After obtaining informed consent and filling in a background questionnaire, the participants were asked to log on to Chreos Tutor and received information about where to find Chreos Tutor and input screens for the two instructional tasks. Following that, the participants were given a pre-test in Chreos Tutor, consisting of two tasks from Client Orders followed by two tasks from Debtor Journal domain, but with no feedback. The time for the pre-test was limited to 10 minutes. After the pre-test, the participants had access to two short instructional videos, explaining how to use the two Chreos screens.

The problem-solving phase (60 minutes long) consisted of 12 problems, presented in the fixed order, starting with 6 Client Orders tasks and followed by 6 tasks from the other domain. The order of task was the same for all participants. The post-test was of similar complexity to the pre-test, but the order of tasks was reversed. After the post-test, the participants completed a questionnaire.

5. Results

A total of 12 volunteers (aged 18 to 29) participated in the study. A number of sessions were offered at various times in order to maximize the number of participants. However, the timing of the study overlapped with assignments the students were taking in other courses, and we ended up with a small number of participants. Due to the haphazard nature of attendance at sessions, it was difficult to ensure an even split between the two conditions. Consequently there were seven participants in the experimental and five participants in the control group. Only one participant noted that English was their first language. Participants were asked to rate their level of computer experience using a Likert-scale ranging from 1 to 7, with 7 indicating that they were very experienced. All participants rated themselves at somewhere between 3 and 5. Three participants had used more than one accounting package prior to the study.

Table 1 presents some statistics from the study. The maximum score on the pre/post-test was 27 (20 marks for the Client Orders and 7 for the Debtor Journal tasks). There were no significant differences on the distributions of pre-test scores between the two groups on the pre- and post-tests, but there was a significant difference on the normalized gains (U = 29.5, p = .048). Three participants from the experimental group obtained perfect scores on the post-test.

Table 1. Pre/post-test performance (standard deviations given in brackets)

	Control (5)	**Experimental (7)**
Pre-test (%)	56.37 (32.79)	40.74 (28.77)
Post-test (%)	93.33 (3.09)	94.71 (7.05)
Normalized Gain	82.78 (4.97)	89.54 (16.16)

We analyzed data collected during the problem-solving phase. Table 2 reports the data for the tasks from the Client Orders domain. The *Attempted tasks* row reports the mean and standard deviation for tasks that participants worked on but have not solved. There were four significant differences found, for the number of attempted and completed problems, time spent and the number of submissions. The experimental group solved a much higher proportion of tasks than their peers from the control group. This is expected because the experimental group participants were not able to move on to a new task until the current task was solved. They also had a higher number of submissions, as they were allowed to submit multiple attempts per task. The last row (*Constraint learnt*) reports the number of constraints the participants learnt during this phase. Those are constraints the participant did not know at the beginning of the phase (i.e. the constraints were violated in initial submissions), but at the end of the phase those constraints were mostly satisfied. There was no significant difference on this measure between the two groups.

Table 2. Statistics about the Client Orders tasks from the problem-solving phase

	Control	Experimental	Significant
Solved tasks	1.6 (.89)	5.57 (.79)	U = 35, p= .003
Attempted tasks	2.8 (1.48)	.14 (0.38)	U = .5, p = .003
Submissions	4.4 (1.14)	29.86 (19.45)	U = 35, p = .003
Time (mins)	24.08 (9.78)	42.64 (5.54)	U = 33, p = .01
Constraints learnt	1.6 (1.95)	3.71 (2.14)	no

Table 3 reports the data for the Debtor Journal tasks. There was no significant difference on the number of completed tasks, but the number of attempted (not solved) tasks is significantly higher for the control group. There is also a significant difference on the number of submissions made, which was expected as the control group participants could only make one submission per task. It is interesting to note that the experimental group required a higher ratio of submissions per solved task for Client Orders tasks than for Debtor Journal tasks.

Table 3. Statistics about the Debtor Journal tasks from the problem-solving phase

	Control	Experimental	Significant
Solved tasks	3.2 (.84)	4.29 (2.06)	no
Attempted tasks	2.8 (.84)	.43 (.54)	U = 0, p = .003
Submissions	6 (0)	9.86 (5.27)	U = 30, p = .048
Time (mins)	19.6 (5.16)	20.39 (5.19)	no
Constraints learnt	0 (0)	.14 (.38)	no

The questionnaire contained questions asking participants to rate the performance of the tutor using Likert-scale ratings ranging from 1 (poor performance) to 7 (excellent performance). Table 4 presents the mean responses from the two groups. The initial two questions asked how well the tutor was able to teach them to complete tasks in the two domains. The ratings the control group participant provided were identical for both domains. The experimental group, on average, rated Chreos Tutor as more effective at teaching Client Orders than the control group, but this view was reversed for Debtor Journals. This suggests that the experimental group found feedback relating to individual errors in submissions less helpful with respect to Debtor Journals. Four participants commented that Debtor Journals were more complex problems and required better explanation than Client Orders.

Table 4. Questionnaire Reponses

Question	Control	Experimental
Tutor’s effectiveness - Client Orders	5.6 (.89)	6.43 (.79)
Tutor’s effectiveness – Debtor Journal	5.6 (.89)	4.71 (1.6)
Satisfied with feedback	4.6 (.55)	5.71 (1.11)
Tutor more effective than printed instructions	5.8 (.84)	5.57 (1.27)
Tutor more effective than instructional video	6 (.71)	5.29 (1.38)
Tutor is interesting to use	6 (1)	6 (1.15)
Confidence – start	4 (1)	3.57 (1.13)
Confidence – end	5 (.71)	5 (1)

The following question was related to the type and level of feedback the tutor provided, followed by two questions asking for a comparison of the tutor to other training options. There was also a question about how interesting they found the tutor. There were no significant differences between the ratings. We also asked students about their confidence in their ability to perform tasks in accounting packages. This question was included in the background questionnaire (reported in the *Confidence –*

start row), and also in the final questionnaire (*Confidence – end*). The participants rated their confidence higher after the session, but there was no significant difference between the groups.

The questionnaire also included some open questions that involved qualitative responses. This feedback was generally positive for both groups. A recurring theme was that the help and feedback provided for the Debtor Journal tasks was not as helpful as that provided for Client Orders. The participants felt that they did not come to understand this type of task as well as they would have liked. The participants also indicated that the process of viewing the tutorial videos needed to be more flexible, by giving them options to pause, rewind or fast forward. Lack of these features made it time consuming to look at parts of the video more than once. Two of the Control Group participants suggested that it would be better if the tutor provided feedback on their particular errors, instead of just showing them the full solution.

In addition to the reported study, we performed an informal case study with a single Chreos user, who used Chreos Tutor in their own time. The user had previous experience with Chreos. The user found Chreos Tutor a more suitable training tool for new users than the existing help documentation, which is more suited to experienced users looking for specific information. Furthermore, instructional videos were insufficient on their own for new users, but the user thought that the videos included in Chreos Tutor were well-paced and acted as a type of introduction to how Chreos input fields work. The user liked the graduated task descriptions, which provided more detail in earlier tasks and just basic information in later tasks. The user commented that the style of feedback was excellent, with graduated levels encouraged people to figure things out for themselves, but the full solution was available for those who were really struggling. When asked to indicate which type of task they felt that the tutor was better at teaching, the response was Debtor Journals. Client Orders are more complicated than Debtor Journals as there is more to learn. On the negative side, the user found that the content of feedback messages with respect to errors in the Client Orders item grid was not as helpful as they could be.

6. Discussion and Conclusions

Embedding ITSs in existing software facilitates on-the-job training in the actual software environment at a time convenient to users. One requirement for 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. We presented Chreos Tutor, a constraint-based tutor embedded into the Chreos business software system.

A study with accounting students showed the beneficial effect of feedback provided by the tutor. The experimental group completed significantly more problems from both domains, and their learning gain measured by the pre- and post-test was significantly higher in comparison to the control group. The experimental group participants were less satisfied with feedback received for the Debtor Journal tasks. However, the experimental group achieved higher scores in the post-test on the Debtor Journal tasks (mean = 91.84, sd = 21.6) in comparison to the Client Order tasks (mean = 87.74, sd = 13.32), with all but one student achieving a perfect score on the Debtor journal tasks in the post-test. The small number of actions necessary to complete a Debtor Journal task makes it reasonably easy to determine what information from the task description corresponds to which input field. Therefore users may learn how to complete task without necessarily understanding the accounting concepts behind the task. The other possible explanation is that the participants were more familiar with the type of tasks included in the Client Orders as these tasks resemble online shopping, while the Debtor Journal domain requires accounting knowledge.

The limitations of our study include the small sample size, due to the timing of the study. Additionally, it would have been preferred to conduct the study with real Chreos users, rather than University students. The experienced Chreos user found Chreos Tutor to be more effective with Debtor Journal tasks, when the majority of the experimental group participants reported the need to improve the way Debtor Journal was taught. Possibly the experienced Chreos user had a deeper understanding of the purpose of Debtor Journals than most of the participants in the study. The feedback messages were composed by someone with an excellent understanding of Debtor Journals. This may have resulted in them being more appropriate for those with a strong accounting background rather than novice users.

Acknowledgements

We thank Wild Software Ltd for their assistance in implementing Chreos Tutor.

References

Amalathas, S., Mitrovic, A., Ravan, S. (2012). Decision-Making Tutor: Providing On-the-Job Training For Oil Palm Plantation Managers. *Research and Practice in Technology-Enhanced Learning*, 7(3), 131-152.

Cheikes, B. A., Geier, M., Hyland, R., Linton, F., Rodi, L., Schaefer, H. P. (1989). Embedded Training for Complex Information Systems. Proc. Intelligent Tutoring Systems, pp. 36-45.

Genesereth, M. R. (1979). The Role of Plans in Automated Consultation. Proc. 6th Int. Joint Conf. on Artificial Intelligence, 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. *Artificial Intelligence in Education*, 7(3-4), 315-347.

Westerfield, G., Mitrovic, A., Billinghurst, M. (2015). Intelligent Augmented Reality Training for Assembly Tasks. *Artificial Intelligence in Education*, 25(1), 157-172.