

Tools for supporting the teaching and learning of data modelling

Pete Thomas
Centre for Research in Computing
Open University
Milton Keynes, UK
p.g.thomas@open.ac.uk

Kevin Waugh
Centre for Research in Computing
Open University
Milton Keynes, UK
k.g.waugh@open.ac.uk

Neil Smith
Centre for Research in Computing
Open University
Milton Keynes, UK
n.smith@open.ac.uk

Abstract: This paper describes three software tools that have been developed to help students learn to produce entity-relationship diagrams (ERDs) in the context of data modelling. The primary tool, called the Exerciser, presents a student with a series of questions and enables the student to produce ERDs that represent the data requirements for the scenarios embedded in the questions. The Exerciser marks each attempt at a diagram and provides several feedback perspectives on the student's answer. The questions, specimen solutions and feedback are built using a tool (the Tutor Tool) which is also described. Both the Exerciser and the Tutor Tool incorporate a purpose-built entity-relationship diagramming tool known as SimpleDraw. The Exerciser has been evaluated and the results of the evaluation are presented. This feedback shows that the tool was found to be very useful and has provided information on which to base a revised version with improved usability.

An Entity-Relationship diagram (ERD) is a graphical notation often used to represent a high-level model of data and their relationships that are to be recorded by a database. An entity represents a discrete type of data item and is named as in: a book, an employee, a sound, or a chemical. A relationship captures how two or more entities are related to one another.

The tools described in this paper are applications of two strands of research: the machine interpretation of diagrams and the automatic marking (grading) of assignment and examination questions [Burstein et al. 2001, Burstein et al. 2003, Haley et al. 2005, Shermis & Burstein 2003, Thomas et al. 2002]. Our investigations of machine interpretation of diagrams have resulted in a framework [Smith et al. 2004] which has been exploited in a tool to automatically mark student answers to assignment and examination questions involving diagrams [Thomas 2003]. Initial trials of the automatic marker have provided confidence that our marking algorithm performs well [Thomas 2003]. While our experiments continue to inform the marking algorithm, we have exploited its capability in a tool, which we have named the *Exerciser*, for practising drawing entity-relationship diagrams (ERDs) in a data modelling context [Waugh et al. 2004].

The Exerciser contains a number of data modelling questions, presented as scenarios, about which students attempt to draw representative ERDs. The tool can mark the diagrams and provide feedback on the accuracy of the diagrams. The tool is intended primarily as a practice and revision tool, to be used following instruction in data modelling techniques.

The *Exerciser* incorporates a drawing tool for drawing ERDs that we have named *SimpleDraw* [Thomas 2004]. *SimpleDraw* has been deliberately designed not to be overly supportive because we were aiming to provide a tool that would enable students to express their own understanding of data modelling – which could be less than perfect – and then provide feedback on this understanding. We did not want a tool that prompted the production of correct diagrams. This fits in with our notion of imprecise diagrams which we employ in our work on machine interpretation of diagrams [Smith et al. 2004]. *SimpleDraw* is also available as a stand-alone tool that is used for capturing student diagrams produced in response to assignment and exam questions. We made use of *SimpleDraw* to convert hand-drawn diagrams into an electronic format (specifically XML) to build up a corpus of student diagrams which have been used in the testing of the automatic marker.

The *Exerciser* works with three distinct items of data: a question (also known as a scenario), an ER diagram, and a set of mappings (associations) between the question and diagram on which feedback is based. The construction of a question and its mappings to a diagram is performed by a third tool called the *Tutor Tool*. The *Tutor Tool* enables the teacher to describe the diagram in terms of the scenario and build mappings which form the basis of the pedagogy in the *Exerciser*.

This paper is structured as follows. Section 2 describes the *Exerciser's* functionality and briefly describes the automatic marking method that underpins it. Section 3 reports on the feedback gained from students on the tool's usefulness and usability. Section 4 describes the *Tutor Tool*. Section 5 summarises our experiences to date and outlines the work we intend to pursue in the future.

The Exerciser Revision Tool

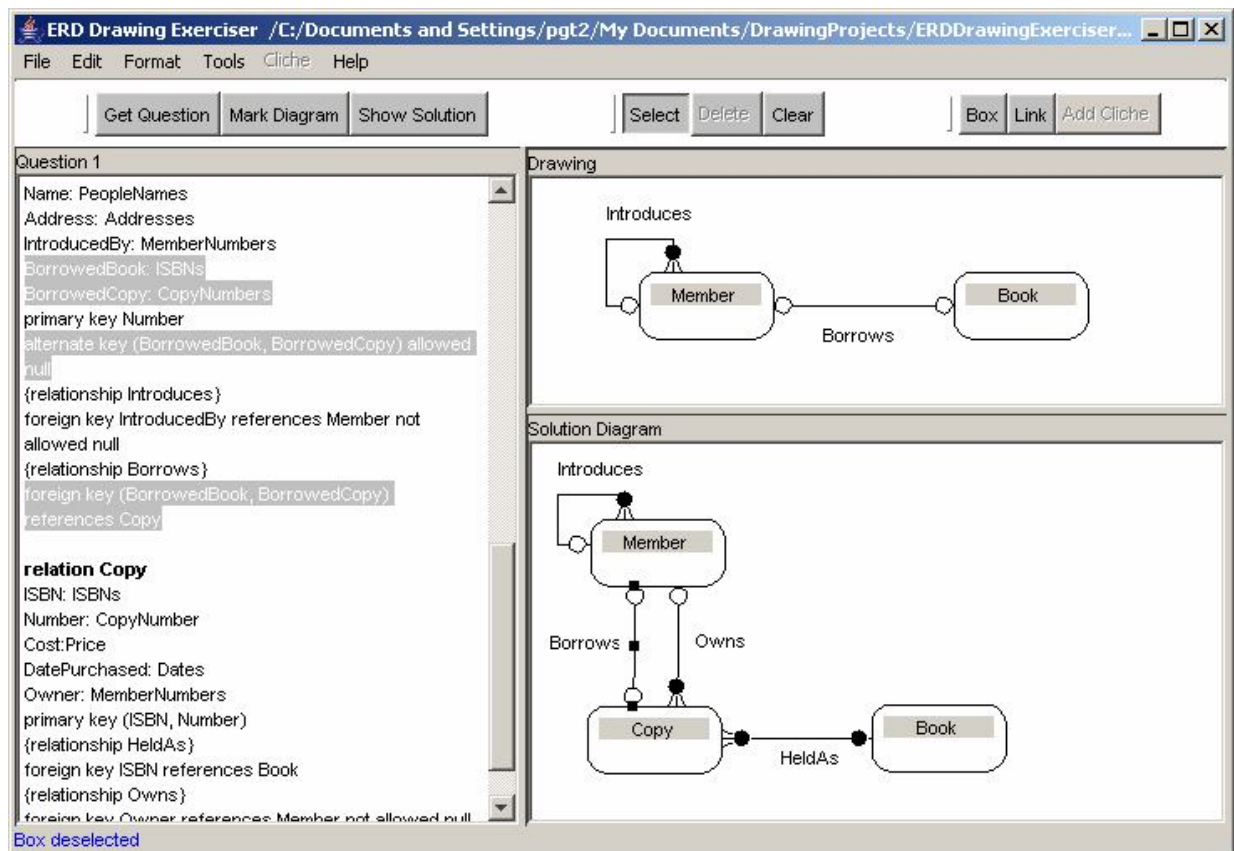


Figure 1: The Exerciser revision Tool

Figure 1 shows the user interface of the *Exerciser* revision tool. When opened, the tool presents the student with a question in the form of a scenario (in the *Question* pane, on the left). The student is expected to draw an appropriate ER-diagram that models the given scenario in the *Drawing* pane (top-right). During the diagram construction phase the diagram in the *Solution Diagram* pane is kept hidden. The solution diagram, a specimen answer to the question, is only revealed when the student presses the *Show Solution* button.

The student is encouraged to attempt the question and then submit their diagram for marking (grading) by the tool. The *Mark Diagram* button triggers the marking process, the immediate result of which is a simple dialog box, an example of which is shown in Figure 2.

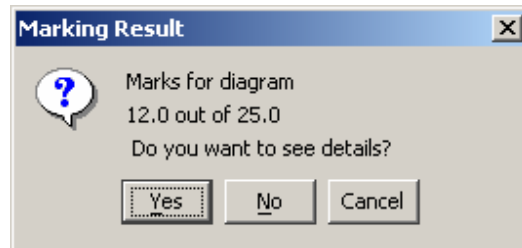


Figure 2: The result of marking a diagram

If required, the student can ask for further details, and would be presented with a feedback window similar to that shown in Figure 3. The feedback window shows the correspondences found by the tool between the relationships in the student's diagram (answer relationships) and those in the specimen solution (solution relationships).

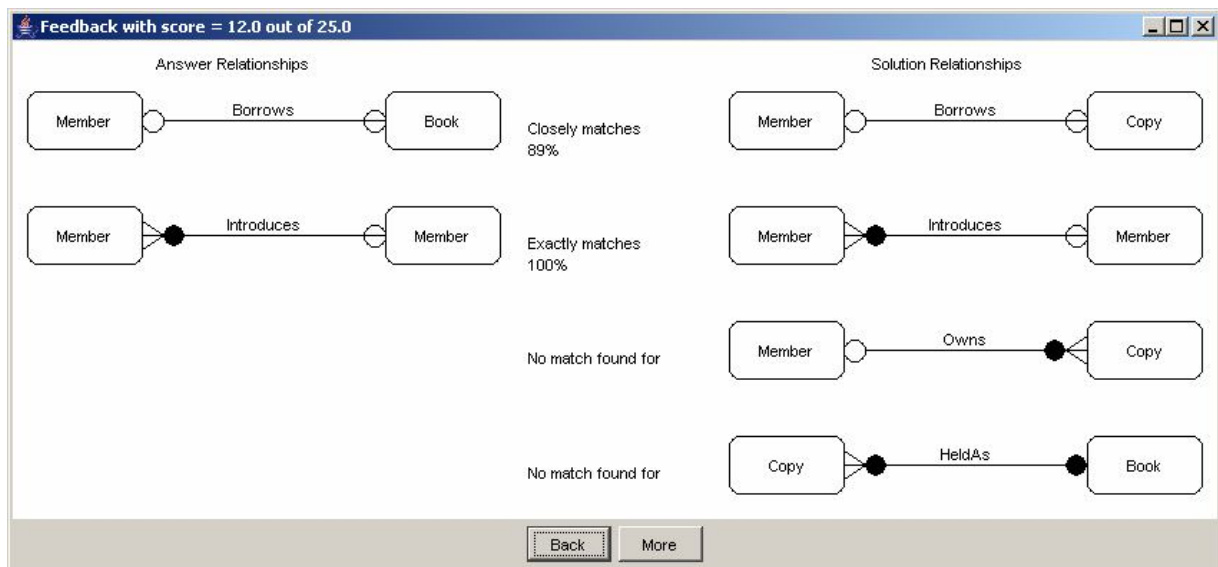


Figure 3: Feedback from Exerciser

There is a further level of feedback in which the student's answer diagram is shown overlaying the specimen solution diagram as illustrated in Figure 4. The overlay indicates which relationships in the student's answer matches the specimen solution (shown by solid lines), which parts are missing (indicated by dotted lines) and which parts are extraneous (lines with mixed dots and dashes – none shown in Figure 4).

Students can revise their diagrams and resubmit them for marking as many times as they like.

Alternatively, the student can reveal the specimen solution and then directly interrogate it. This is achieved by selecting an element of the solution diagram. The tool will then highlight those parts of the model which give rise to

the selected diagram element (this is shown in Figure 1, where the selected *Borrows* relationship leads to the highlighting of the data relating to the relationship).

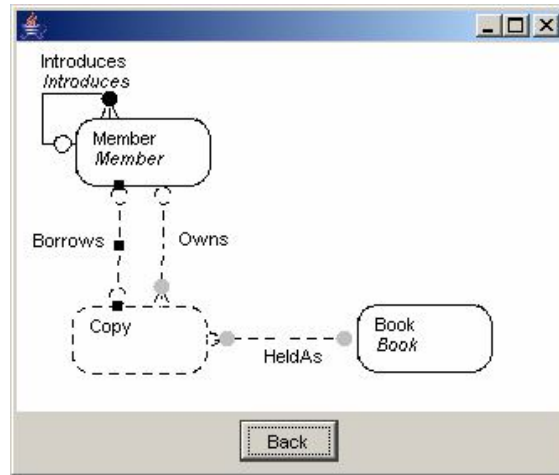


Figure 4: Further feedback: overlays

Right-clicking on a diagram element reveals a *Properties* window which gives more details of the diagram element and its derivation (see Figure 5).

The 'Link Properties' dialog box shows the following details for the 'HeldAs' relationship:

Copy		Book	
Crows Feet	Yes	No	No
Blob	Closed	Closed	Closed

In the relational model, the Copy relation has a relationship named HeldAs with a foreign key that references the relation Book. This implies that there is a one-to-many relationship from Book to Copy.

Figure 5: A properties window

The tool maintains a number of questions (10 in the current implementation) which can be selected via the *Get Question* button (Figure 1).

Automatic Marking

The method we use for automatically grading (marking) a diagram is based on a framework for diagram interpretation described in [Smith et al. 2004]. The approach begins by decomposing a diagram into its component parts called minimal meaningful units (MMUs) – the smallest elements of a diagram that has meaning in the domain being modelled. In ERDs, the MMUs are entity types and associations (there are two associations – relationships, represented by a line joining two entity types, and supertype-subtype relationships, represented by drawing an entity

type (the subtype) inside another entity type (the supertype). The MMUs in a student-drawn diagram are compared with the MMUs in a specimen solution diagram (using a similarity measure based on the properties of the different MMU types). The measures of similarity are used to match the MMUs in the student diagram with those in the specimen solution. A mark and the feedback are derived from a marking scheme applied to the matched MMUs.

This basic scheme is modified in two important ways. First, the comparisons are based largely on the names used for the entity types and relationships and we use elements of natural language processing [Manning & Schütze 2002] to support this process (for identifying synonyms and misspellings, for example) [Thomas et al. 2005]. Second, there are occasions when it is useful to aggregate the MMUs into larger elements to identify equivalent sub-structures in diagrams [Thomas et al. 2006].

The *Exerciser* was originally devised as a way of validating the automatic marking method in a low-stakes application prior to certifying it for assessment purposes. The development of the automatic marker is an on-going research project.

Evaluation of the Revision Tool

The *Exerciser* revision tool was given to volunteer students studying a database course in which ER data modelling is taught. Twenty-eight students responded to a questionnaire designed to elicit opinions of the usefulness and usability of the tool.

Usefulness

The tool had 10 reasonably demanding questions for students to attempt (the questions were of an equivalent standard to those normally found in assignment questions). On average, the students attempted over 4.4 questions each, which is certainly enough to gain a good impression of the usefulness and usability of the tool. The main reasons given by students for not attempting more questions were either that they had completed enough questions to satisfy themselves that they understood ER-diagramming adequately or that they had insufficient time with several students citing the need to complete an imminent assignment. Several students said that they would be returning to the tool during their revision. We can conclude, therefore, that the responses to the questionnaire were based on a sufficiently good understanding of the tool and its capabilities.

All the students who returned the questionnaire said that the tool was helpful in their studies and 6 volunteered that the tool was *very* helpful. All students said that they would recommend the tool to other students with 6 volunteering that they would *definitely* recommend the tool. Most students provided additional positive comments on the usefulness of the tool. Typical responses were:

“I found it a great learning experience.”

“I love it – being able to see where parts of the diagram come from is magic.”

“I particularly liked the feature in ‘Mark diagram’ that gave a % of match to the solution – it gave me hope!”

“Good tool, which has been and still will be, helpful to me!”

“I enjoyed using the software and thought that it was very helpful in enhancing my understanding of E-R diagrams.”

“I liked the explanation ... describing the origin of the selected element.”

“It was the final piece of the jigsaw for me. It was a good opportunity to test understanding of the theory of the course.”

Usability

The average time to become familiar with the tool was just over 13 minutes with most students quoting 10 minutes. We can consider this to be a good result because students were provided only with a brief description of the tool's features (little more than appears in this paper) and had to rely on the tool's help facility for some details; 17 students used the help facility but 9 said that they found the tool sufficiently intuitive that they did not access the help.

21 students said that they found the tool easy to use, 3 felt that it was not easy to use and 4 did not respond to the question. The responses indicated that the tool contained a number of faults (software errors) as well as some annoying behaviours (design faults). In particular, many students found that they accidentally created multiple unwanted boxes which they found difficult to delete (this was intended as a 'feature' for enabling boxes to be drawn quickly, with a single mouse click, which was clearly too effective!). When coupled with the fact that the instructions for the tool (accessed via the tool's help menu) were both incorrect and incomplete in places, it is remarkable that these students did not give up using the tool. No student reported that the issues with usability detracted from the usefulness of the tool.

The user interface shown in Figure 1 is an evolution of that used in the tool that was evaluated and described in [17] and is based on the feedback gained in this evaluation. It is currently being re-evaluated as students are using the new tool during their revision in the most recent course presentation. Early indications are that the *Exerciser* tool is even easier to use.

Conclusion

We conclude that the students who responded found the revision tool to be very useful despite its usability deficiencies. There was little criticism of the main functionality of the tool and indeed many students said that they wished such a tool had been available as part of the course materials. The overwhelming positive reaction to the tool together with the fact that all respondents said that they would recommend the tool to other students gives us confidence that there is a need for such a tool and that the current implementation is both appropriate and effective.

The Exercise Builder (Tutor Tool)

For the original trials of the *Exerciser*, the diagrams were built using the *SimpleDraw* diagramming tool but the scenarios and mappings were hand-crafted. That is, the data required by the *Exerciser* were contained in text files which were constructed by hand using a text editor. This meant that the specimen solutions were time consuming to build and difficult to maintain. We therefore built a tool designed for building specimen solutions that we designated the *Tutor Tool*, see Figure 6.

The tool maintains two window panes: on the left, the tabbed *Question* pane is used for constructing scenarios; on the right, the *Diagram* pane is used for constructing diagrams.

Both windows have their own tool bar containing buttons for editing their content. In the case of the scenarios, there are elementary formatting facilities (for boldening, italicising and underlining selected portions of the text – limited by the needs of the *Exerciser*), but with a *Format* menu providing some support for future expansion. The *Question* pane is effectively an embedded version of *SimpleDraw*.

The primary function of the *Tutor Tool* is to associate or, map, elements in the diagram to elements in the question. This is achieved by selecting a single element of the diagram and a contiguous portion (substring) of the scenario and clicking the green *Map* button. Multiple selections from the scenario can be associated with the same diagram element by repeated use of the *Map* function.

The *View All* button highlights all the selections in the scenario that have been mapped to the selected diagram element. As a quick visual indication that an element of the diagram has been mapped to at least one element of the scenario, the marker, (m), is appended to the name of the element in the *Diagram* pane (shown for all the named elements in Figure 6).

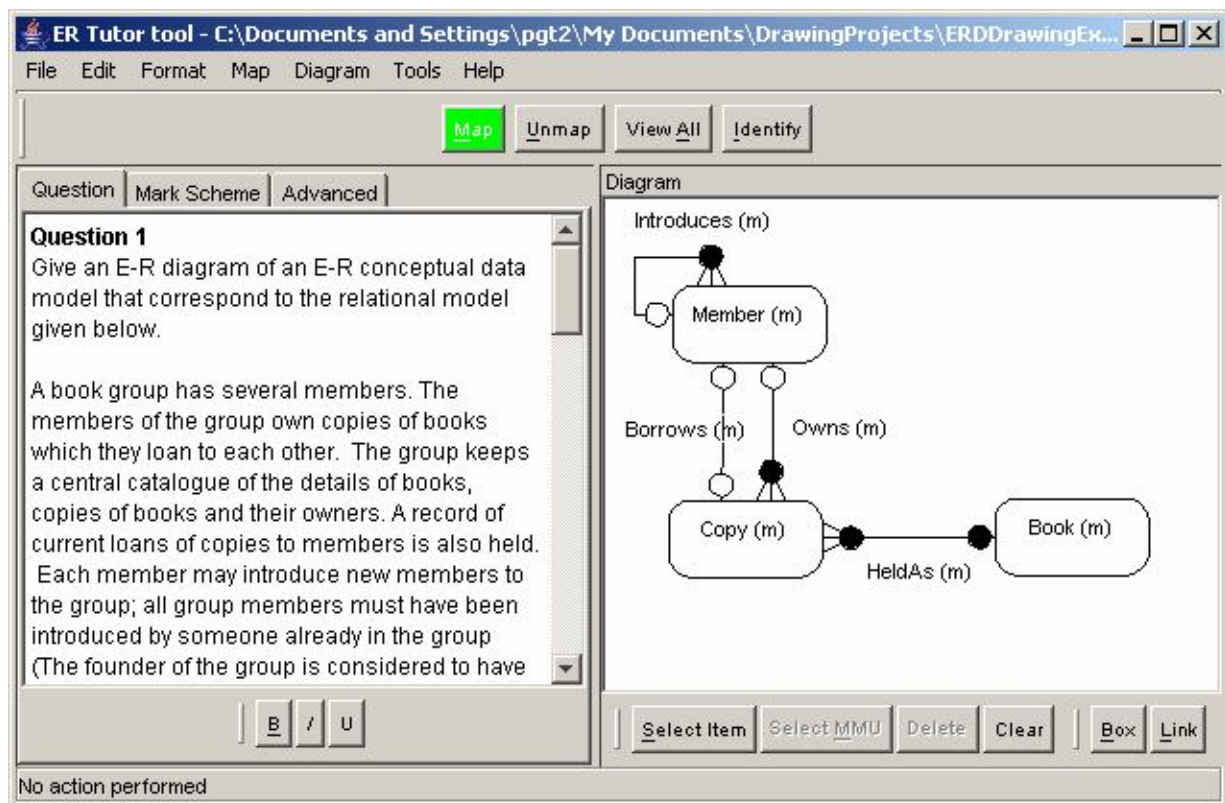


Figure 6: The Tutor Tool

Mappings between an element of the diagram and element(s) of the scenario can be deleted by clicking the *Unmap* button. In the current implementation, the *Unmap* function removes all mappings for the selected diagram element.

If the user wishes to know whether an element of the scenario has been mapped to an element of the diagram, they first select the portion of text in the scenario and then click on the *Identify* button. The mapped elements of the diagram are highlighted.

The main advantage of the *Tutor Tool* is that it maintains the mappings between the scenario and the diagram while the scenario and diagram are being edited.

Once the user is satisfied with the scenario, diagram and specified mappings, the complete specimen solution can be saved in an XML format that can be read by the *Exerciser* tool.

An important feature of the *Exerciser* revision tool is its ability to mark (grade) a student attempt at drawing an ERD that models the given scenario. As illustrated above, the marking feature provides feedback in graphical form. To support this feature, the *Tutor Tool* enables the user to specify a mark scheme as illustrated in Figure 8.

The tool supports three different ways of setting a mark scheme (as indicated by the three radio buttons on the left-hand side of Figure 2). In the first method, known as ‘whole diagram marking’, a single mark for a correct diagram is specified. In the second method, the major elements of the diagram (entities and relationships) are given individual marks. This can be done in two ways: either every entity has the same mark and every relationship has the same mark (as illustrated in Figure 8) or each individual entity and relationship has its own mark (set via the *Advanced* button). The third method enables the user to allocate marks to the adornments within the diagram.

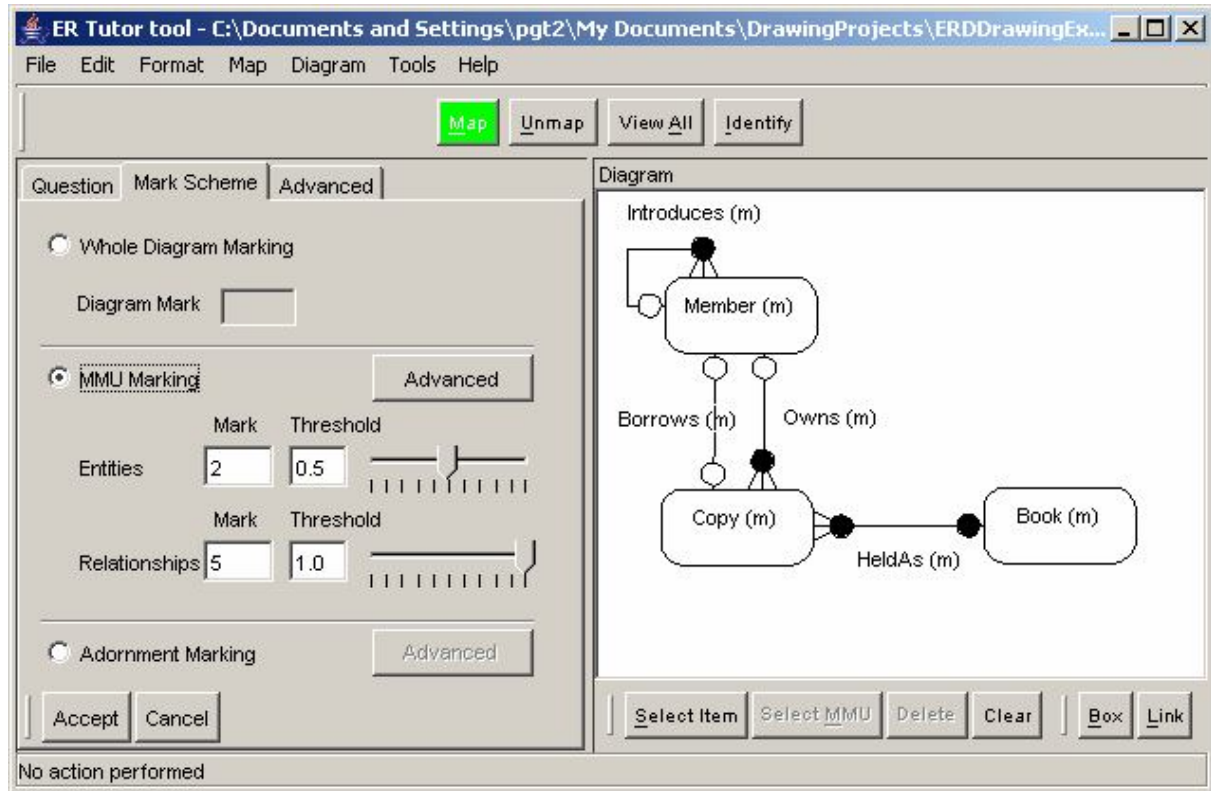


Figure 8: Setting a mark scheme

In addition, the tool allows the user to set a threshold on the level of similarity (a value between 0 and 1) needed to be reached between two elements if they are to be considered equivalent. For example, if the automatic marker decides that a particular relationship in the student answer is similar to a relationship in the specimen solution at the level of 0.5 (only about 50% similar) then the student's relationship would not gain any marks if the threshold had been set above 0.5.

The *Tutor Tool* has one further feature: the ability to create a table of synonyms. In this work, students can invent their own names for entity types and relationships and the automatic marking needs to be able to relate student chosen names with names in the specimen solution. In so doing, the marker needs to be aware of synonyms – either words that are synonymous in everyday speech or synonyms which are specific to the domain being modelled. The synonym building tool is intended for capturing the latter – a task that is most easily undertaken when creating the question scenario.

The *Tutor Tool* has been described in terms of its primary use – devising questions and specimen solutions for the *Exerciser* tool. However, it has another use: for building specimen solutions for use by an automatic marker (embedded in the *Exerciser* tool but also available as a stand-alone tool for automatically marking diagrams [Thomas et al. 2005]). Therefore, the tool has been designed to allow specimen solutions to be created and stored independently of the question.

Summary and Future Work

In this paper, we have described three software tools for supporting the teaching of data modelling. The *Exerciser* is a student revision tool for enabling students to practice drawing entity-relationship diagrams which has been evaluated and modified to produce a useful and usable tool. The *Tutor Tool* enables teachers to build the data required by the *Exerciser* in a natural manner. The third tool is an ER-diagram drawing tool which is embedded within both the *Exerciser* and *Tutor Tool* but is also available as a stand-alone application.

The ultimate aim of the work presented here is to embed the *Exerciser* into a new database course. The immediate need is to produce a new set of questions (using the tutor tool) which are tailored to the new course. The *Exerciser*, and the revised questions, will then be evaluated by students on the new course.

The underlying technology of the automatic marker is being exploited in other tools, in particular, a tool to support humans in grading diagrams. We have found evidence that the structure of a student diagram can lead human markers to misinterpret it and award lower marks than they should. By using the matching capability of the marker, it is possible to redraw a student diagram, without any loss of information, in a form that more closely resembles the specimen solution. The human is then more able to award an appropriate mark. This marker's support tool is currently in prototype stage.

The tools presented here are an application of our work on diagram understanding. One of the aims of this research is to produce a software application that can be used in summative assessment. The *Exerciser* revision tool was built as a mechanism for gaining feedback on the effectiveness of the marking algorithm. The feedback will enable us to provide better automatic marking tools and improved feedback mechanisms for the *Exerciser*, an area of wider interest [Batmaz & Hinde 2006, Higgins & Bligh 2006, Hoggarth & Lockyer 1998].

References

- Anderson, M., & McCartney, R. (2003). Diagram processing: Computing with Diagrams. *Artificial Intelligence* **145** (1-2), 181-226.
- Batmaz, F. & Hinde, C.J. (2006). A Diagram Drawing Tool for Semi-automatic assessment of Conceptual Database Diagrams. In *Proceedings of the 10th Annual International Conference in Computer Assisted Assessment*. Loughborough University, Loughborough, UK, July 2006, 68-81.
- Burstein, J., Chodorow, M. & Leacock, C. (2003). Criterion SM Online Essay Evaluation: An Application for Automated Evaluation of Student Essays. In *Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*, Acapulco, Mexico. August 2003.
- Chok, S.S. & Marriott, K. (1995). Parsing visual languages. In *Proceedings of the Eighteenth Australian Computer Science Conference*, Australian Computer Science Communications, **17**, 90-98.
- Haley, Debra Trusso, Pete Thomas, Anne De Roeck, & Marian Petre. (2005). A Research Taxonomy for Latent Semantic Analysis-Based Educational Applications. In G. Angelova, K. Bontcheva, R. Mitkov, N. Nicolov & N. Nikolov (Eds.), *International Conference on Recent Advances in Natural Language Processing'05* Borovets, Bulgaria. 575-579.
- Higgins, C. A. & Bligh, B. (2006). Formative Computer Based Assessment in Diagram Based Domains. In *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006)*, June 26-28, Bologna, Italy, 98-102.
- Hoggarth, G. & Lockyer, M. (1998). An Automated Student Diagram Assessment System, *ACM SIGCSE Bulletin*, **30**(3), 122-124.
- Iizuka, K., Tanaka, J. & Shizuki, B. (2001). Describing a drawing editor by using constraint multiset grammars. In *Proceedings of the Sixth International Symposium on the Future of Software Technology (ISFST 2001)*, Zhengzhou, China. November, 2001. www.iplab.is.tsukuba.ac.jp/paper/international/iizukia-isfst2001.pdf (accessed 02/06/04)
- Manning, C.D. & Schütze, H. (2002). *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts. ISBN 0-262-13360-1.
- Marriott, K., Meyer, B., Anderson, M., Cheng, P. & Haarslev, V. (2000). Non-standard logics for diagram interpretation. In *Proceedings of Diagrams 2000, International Conference*, Edinburgh, ISBN 3-540-67915-4-42-57.
- Shermis, M.D, & Burstein, J.C. (2003) (Eds.) *Automated Essay Scoring: a cross-disciplinary approach*. Lawrence Erlbaum Associates, Mahwah, NJ, USA. ISBN 0-8058-3973-9.

Smith, N, Thomas, P.G. & Waugh, K. (2004). Interpreting Imprecise Diagrams. *In Proceedings of the Third International Conference in the Theory and Application of Diagrams*. March 22-24, Cambridge, UK. Alan Blackwell, Kim Marriott, Atsushi Shimojima (Eds) Springer Lecture Notes in Computer Science, 2980, 239-241. ISBN 3-540-21268-X.

Thomas, P.G., Price, B., Paine, & C. Richards, M. (2002). Remote Electronic examinations: an architecture for their production, presentation and grading. *British Journal of Educational Technology (BJET)*, **33** (5) 539-552.

Thomas, P.G. (2003). Evaluation of Electronic Marking of Examinations, *In Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2003)*, Thessaloniki, Greece, 50-54.

Thomas, P.G. (2004). Drawing Diagrams in an Online Exam, *In Proceedings of the 8th Annual International Conference in Computer Assisted Assessment*. Loughborough University, Loughborough, UK, 403-413.

Thomas, P.G., Waugh, K., & Smith, N. (2005). Experiments in the Automatic marking of E-R Diagrams. *In Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2005)*, Monte de Caparica, Portugal, 158-162.

Thomas, P.G., Waugh, K., & Smith, N. (2006). Using Patterns in the Automatic Marking of ER-Diagrams. *In Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE 2006)*, Bologna, Italy, 403-413.

Tsintsifas A. (2002). *A Framework for the Computer Based Assessment of Diagram-Based Coursework*, Ph.D. Thesis, Computer Science Department, University of Nottingham, UK.

Waugh, K.G., Thomas, P.G., & Smith, N. (2004). Toward the Automated Assessment of Entity-Relationship Diagrams. *In Proceedings of the 2nd LTSN-ICS Teaching, Learning and Assessment in Databases Workshop*, Edinburgh, Scotland. <http://www.ics.ltsn.ac.uk/pub/databases04/index.html> (accessed 15/01/06)

Acknowledgements

Part of this work was undertaken while one of the authors held a HEFCE funded Teaching Fellowship of the Centre for Open Learning in Mathematics, Science, Computing and Technology at the Open University.