

ALT-C 2007 Research Paper

Authors: Pete Thomas
Neil Smith
Kevin Waugh

Address for correspondence: Dr P.G.Thomas
Computing Department
Faculty of Mathematics and Computing
The Open University
Milton Keynes
MK7 6AA
p.g.thomas@open.ac.uk

Dr N. Smith
Computing Department
Faculty of Mathematics and Computing
The Open University
Milton Keynes
MK7 6AA
n.smith@open.ac.uk

Dr K.G.Waugh
Computing Department
Faculty of Mathematics and Computing
The Open University
Milton Keynes
MK7 6AA
k.g.waugh@open.ac.uk

Learning and automatically assessing graph-based diagrams

Abstract

To date there has been very little work on the machine understanding of imprecise diagrams – diagrams drawn by students in response to assessment questions. While there have been successful attempts at assessing text (essays) automatically, little success with diagrams has been reported. In this paper, we explain an approach to the automatic interpretation of graph-based diagrams based on a 5-stage framework. The paper reports on some experiments in automatically grading student diagrams produced under examination conditions which show good agreement with the performance of human markers. The paper also describes how the automatic marking algorithm is being used in a variety of teaching and learning tools.

Introduction

In this paper we shall describe a number of software tools designed for learning and assessing graph-based diagrams. Typical examples are entity-relationship diagrams, UML diagrams, biological flow diagrams and chemical structure diagrams. In the work described here, we have concentrated on entity-relationship diagrams (ERDs) used in data modelling. ERDs are a simple but effective mechanism for eliciting, analysing, recording and communicating information about data that a business enterprise might store in a database. Typically, constructing ERDs is a skill normally taught in a database course but is useful in other areas such as requirements engineering.

It is common to see diagramming tools used in the teaching of ERDs, but these tend to be specifically designed to support the construction of correct diagrams by professional data modellers. In a teaching environment, we would prefer to have tools that are less constraining and which allow students to express their understanding – or lack of it – and have the tools give feedback on the errors being made. That is, the tools would deal with imprecise diagrams. Such tools would enable a student to have multiple attempts at a problem and to investigate the problem in a variety of ways. Tools that provide this kind of feedback could also grade student work and one can imagine such tools being an integral part of a computer aided assessment system, particularly if they could be applied to a wider domain than data modelling.

Our interest in assessing student attempts at drawing ERDs stemmed initially from our attempts to mark (grade) online examinations, albeit in a formative environment (Thomas et al., 2002, Thomas, 2003, 2004). Examination questions in Computing often elicit answers that include diagrams. Therefore, having a drawing tool attached to an online exam paper would be helpful. One of the advantages of an online exam is that answers are captured in electronic form and potentially can be graded automatically. While there are several systems being developed for grading textual material (Burnstein et al., 2003, Haley et al., 2005) and there is a considerable literature for describing diagrams (see, for example, (Anderson & McCartney, 2003), (Chock & Marriott, 1995), (Kniverton, 1996) and (Marriott et al, 1998)) there is very little work on grading diagrams. Tsintsfas (2002) has produced a framework for the assessment of diagram-based coursework which has fed into an ERD tool within the CourseMarker CBA system (Higgins & Bligh, 2006) and Batmaz and Hinde (2006) have investigated a semi-automatic marking system. We decided to investigate what might be possible.

In the following sections we discuss the general problem of diagram interpretation, our approach to automatically marking a diagram, and the tools we have built to exploit this technology for learning and assessment. The final section discusses how we intend to take this work forward.

Diagram interpretation

Figure 1 shows a typical ERD. It is the specimen solution to a data modelling problem and has been laid out in a form that a human marker can readily comprehend. Such a diagram is feature-based in the sense that it is composed of several types of features: boxes with names inside, lines with names beside them, circles (either open or filled) and, at the ends of some lines, ‘crowsfeet’. Each of these features has meaning within the context of the diagram. For example, a box represents an entity type in which the name serves to label the type of entity being described. A line with a crow’sfoot at one end represents a one-to-many relationship between the entity types at either end of the line. However, some features, if they were to appear on their own would not convey meaning. An open circle on its own is still an open circle but carries no meaning in the context of an ERD, but if it appears at the end of a line joining two entity types, it represents an optional participation. It turns out to be useful to identify diagram features (or small combinations of features) that can appear on their own and carry meaning. In an ERD, there are three such structures: an individual named box represents an entity type, a named line joining two entity types represents a relationship in which the name labels the type of relationship, and a box wholly containing another box represents a supertype-subtype relationship (an example can be found in Figure 5). We refer to instances of these three structures as minimal meaningful units (MMUs) because, if any part of the structure is removed, it ceases to convey meaning in this domain. Generally speaking, MMUs can be aggregated to form larger structures with more meaning. But since these diagrams are no longer minimal they are called meaningful units (MUs). Thus, the diagram in Figure 1 is an MU, and it contains 6 entity type MMUs and 7 relationship MMUs.

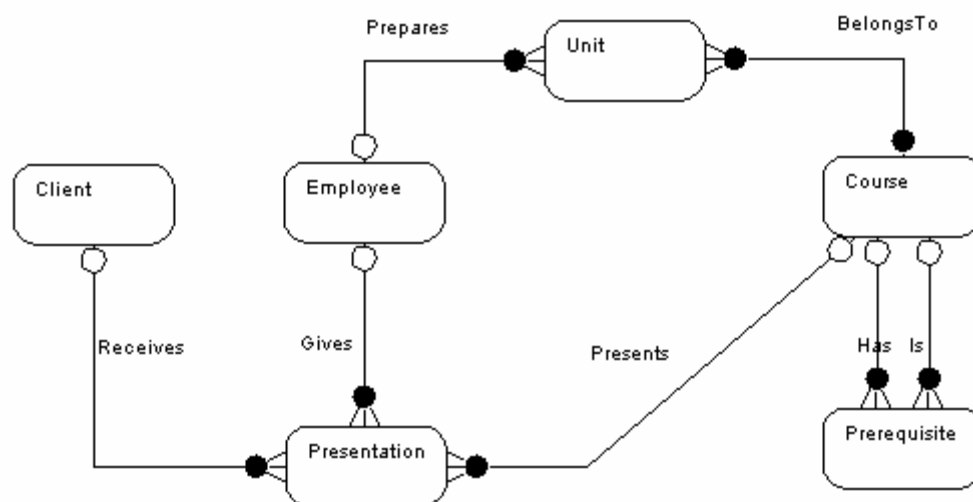


Figure 1 An entity-relationship diagram

It is important to check these margins even if you use this Word template, because they might have been overwritten by your local settings.

In the domain of data modelling, there are certain small diagrams (MUs) that have useful properties. For example, Figure 2 illustrates an equivalence between two MUs. The MU at the

top consists of two one-to-many relationships with specific participations. This aggregate MU can be interpreted – in this domain – as equivalent to the MU (actually an MMU) shown below it.

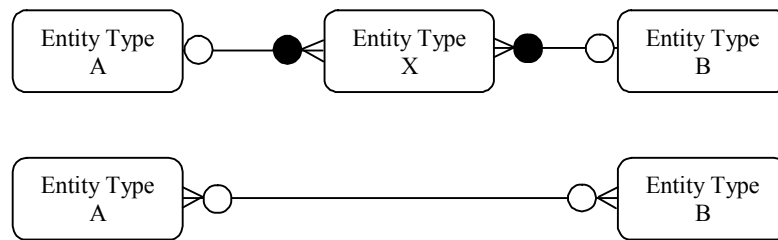


Figure 2 Equivalent MUs

However, in the area of assessment, imprecise diagrams frequently occur. An imprecise diagram is one where required features are either malformed or missing, or extraneous features are included (Smith et al., 2004).

These observations, among others, have led us to the five stage framework for interpreting imprecise diagrams shown in Figure 3.

1. segmentation	}	General diagram knowledge	Convert raster-based images to general diagram features
2. assimilation			
3. identification	}	Domain specific knowledge	Identify MMUs, combine into larger MUs and interpret
4. aggregation			
5. interpretation			

Figure 3 The 5-stage framework for imprecise diagram interpretation

The first two stages, segmentation and assimilation, use general diagram knowledge to identify basic segments such as lines and arcs to produce features such as boxes and circles. However, our primary interest is in stages 3 to 5 where domain specific knowledge is required to interpret the diagram. The identification stage takes the diagram features from stage 2 and identifies MMUs. The aggregation phase combines MMUs to form useful MUs, and finally stage 5 takes the collection of MUs and interprets their meaning. In the work reported here, we say that we have successfully interpreted a diagram when we have successfully graded the diagram and provided meaningful feedback.

Automatic marking

Similarity measures

Figure 4 shows an example of an ERD which was drawn by a student in answer to a problem to which Figure 1 is the specimen solution. There are clear similarities between the two diagrams, but there are also some differences. In our environment, the assimilation stage is performed by a drawing tool which allows students to create boxes, lines, circles and so on to produce a diagram that purports to be an ER diagram. The identification stage converts the diagram features into MMUs. In the case of Figure 4, 6 entity types and 5 relationships can be identified.

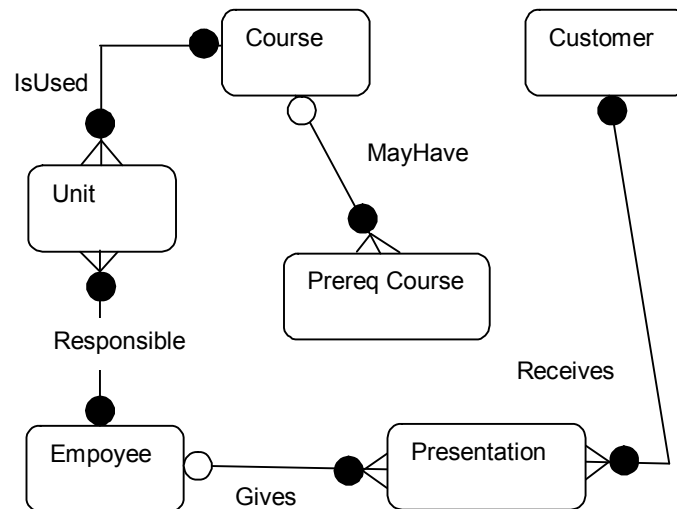


Figure 4 A student drawn ERD

The interpretation stage compares the student answer diagram with the specimen solution diagram by comparing the sets of MMUs in the two diagrams. (Our current automatic marker does not include an aggregation stage, although one is needed as we discuss later.) Thus, the entity types in the student diagram are compared with the entity types in the specimen solution and, likewise, the two sets of relationships are compared. The aim of the comparisons is to determine the best possible match between the MMUs in the student answer diagram and the same types of MMU in the specimen solution.

The comparison is performed using similarity measures. That is, each MMU of a given type in one diagram is compared with every MMU of the same type in the other diagram and each time a measure of their similarity is computed. A similarity measure is a real number in the range [0..1] with 1 representing an exact match. The aim is to find the best possible match between the two diagrams by maximising the sum of the similarities for each MMU.

The advantage of computing a numeric similarity measure is that it enables the graceful handling of imprecision in a student's diagram. Currently we use edit distance between the names as the basis of the similarity measure. This simple scheme has the advantage that simple spelling errors can be taken into account ('Employee' is very similar to 'Emplooyee'). However, the simple scheme really only works well when the student uses simple names that are close to those used in the specimen solution. Dealing with more complex naming requires synonyms to be handled effectively. In addition, the names assigned to entity and relationship types tend to be 'composite' – formed from several words – to provide slightly more meaning that would be conveyed by a single word. For example, 'Prereq Course', 'IsUsed' and 'MayHave' in Figure 4.

Therefore, we have modelled entity names as noun phrases and relationship names as verb phrases. The subsequent processing attempts to extract the noun from the noun phrase and the verb from the verb phrase and bases the similarity of names primarily (but not exclusively) on the nouns and verbs. (Stop words are discarded.)

Further processing attempts to deal with abbreviations, which are common in this area especially when students are under time pressure as in an examination. Hyphenation that does not follow normal conventions is also popular! The final form of a name is a list of words each one of which is stemmed to deal with differences in parts of speech (receives, received and receiving would all be considered synonymous).

Since an aim of our research is to produce usable systems that can be incorporated into teaching and learning tools which execute on typical PCs with reasonable response, we have to balance the extent of the natural language processing with performance.

Synonyms

Of particular importance in accurate automatic marking is the detection of synonyms. Synonyms in every-day speech can be dealt with by conventional look-up techniques, and there are likely to be well-known domain specific synonyms, that can be handled similarly. However, it is possible to detect other names used synonymously. For example, the name 'Prereq Course' is an example of a hyponym. If the specimen solution used the word 'Course', then 'Prereq Course' can be assumed to be a synonym because it is a type of 'Course'. This would mean, in Figure 4, that 'Prereq Course' would be considered to be similar to 'Course' and also similar to 'Prerequisite'. This is not necessarily a problem provided that the similarities of 'Prereq Course' and 'Prerequisite' to 'Course' are less than 1 to allow the similarity of the occurrence of 'Course' in both diagrams to be bigger, at precisely 1. Recall that the aim is to determine the best *overall* match between the MMUs of the two diagrams; therefore, provided the matching of words that might possibly be similar does not interfere with the matching of truly synonymous words, the scheme works well.

Another, domain-specific, source of possible synonymy is where there is a one-to-one relationship between two entity types and one of those entity types is not related to any other entity type. In this case, the two entity type labels can be viewed as synonyms.

Such rules have to be applied carefully. In diagrams where supertype-subtype relationships are present, hyponymy is to be expected and it is necessary to match supertype with supertype and subtype with subtype and not allow the obvious similarity between supertype and subtype to confuse the matching – the supertype name needs to be distinguishable from the subtype name.

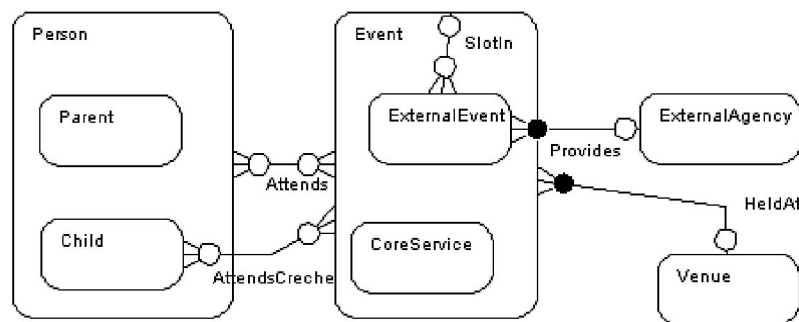


Figure 5 An ERD containing supertype-subtype relationships

The similarity measures incorporate a number of weights and thresholds. The weights are used to change the significance of diagram features to reflect their importance in different domains. The thresholds are used to avoid matching MMUs where the evidence (similarity) is limited. In some cases, the level of a threshold has been determined by experiment.

Marking schemes

Having found the best possible match between sets of MMUs, a marking scheme can be applied. In general, there can be a wide variety of marking schemes, often depending upon the concepts being assessed at a particular stage of a course. This makes marking schemes quite difficult to construct. One approach would be to provide a scripting language for specifying marking schemes but it is doubtful whether it would be used in practice. Therefore, we devised

a parameterised marking algorithm which, while not comprehensive, is adequate for marking many uses of ERDs in different learning contexts.

Evaluation and results

In our early experiments with marking ERDs (Thomas et al., 2005, Waugh et al., 2004) we gained experience with relatively small sets of diagrams. These experiments gave us confidence in the approach but we required a much larger corpus of diagrams to provide more convincing evidence. Therefore, we set about constructing a sizeable corpus of student drawn ERDs. We gathered almost 600 diagrams drawn in a real, invigilated examination (the diagram shown in Figure 1 is the specimen solution to the question posed). The diagrams were graded by a group of human markers whose work was moderated (checked for consistency and accuracy by an independent marker) to provide a corpus of diagrams with marks which were felt to be accurate reflections of the diagram content.

We arbitrarily divided the corpus into two sets of diagrams. One set of almost 200 diagrams was used in the development of the marking algorithm, for detecting bugs and setting some thresholds (in NLP, such sets are often called training sets). The remaining diagrams (394) were used to test the accuracy of the automatic marker when compared to the human markers. Table 1 shows the results of this experiment. The diagrams were marked out of 7 and rounded to the nearest half mark.

Table 1 Differences between the auto marker and humans (1)

Difference	0	0.5	1.0	1.5	2.0	2.5
Number	271	89	32	1	0	1
%	68.8	22.3	8.1	0.26	0	0.26
Cum. %	68.8	91.4	99.5	99.8	99.8	100

The results show that the automatic marker agrees exactly with the human markers in almost 69% of cases. However, given the variation often present in human marking (Newstead & Dennis, 1994) we feel that the percentage of cases which differ by no more than half a mark is a better reflection of the automatic marker's accuracy: over 91% are in this category. The one outlier, which differs by 2.5 marks from the human mark, contains an error in which the student had named two entity types identically. The human marker had given 'the benefit of the doubt' and awarded credit for something the automatic marker had viewed as a significant error. Which is the better approach is a matter for debate.

This experiment, while giving an encouraging result, can be criticised on the basis that only two of the three ERD MMUs are tested. Therefore, we applied the automatic marker to a more complex ERD containing supertype-subtype relationships (again obtained from a real, invigilated examination; Figure 5 shows the specimen solution). At the time of writing we have a corpus of 30 such diagrams with moderated human marks. Table 2 shows the latest results.

Table 2 Differences between the auto marker and humans (2)

Difference	0	0.5	1.0	1.5	2.0	>2.0
Number	11	18	0	1	0	0
%	36.7	60.0	0	3.3	0	0
Cum. %	36.7	96.7	96.7	100	100	100

While the exact matches are not as good as in the first experiment, the 96.7% of cases differing by no more than 0.5 is better. However, the students' answers tended to be quite poor and so the range of human marks is much smaller than in the first experiment.

It is also interesting to compare the behaviour of the automatic marker in dealing with the two kinds of relationship. The recognition of supertype-subtype relationships (on the basis of marks awarded) is better than for ordinary relationships as shown in Table 3.

Table 3 Comparing the auto marker performance on two MMU types (%)

Difference	0	0.5	1.0	1.5	2.0	>2.0
Relationship	50.0	46.7	0	3.3	0	0
Supertypes	86.7	13.3	0	0	0	0

Armed with this information, we felt confident that the technology could be incorporated into a number of software tools to support learning and assessment, particularly in a formative environment.

Tools for learning

Revision tool

The revision tool enables a student to draw an ERD in response to a textual problem. Once satisfied with their diagram, the student requests the tool to mark the diagram and provide feedback. The user interface for the tool, after the student has drawn their diagram (top drawing) and asked to see the specimen solution (bottom drawing) is shown in Figure 6.

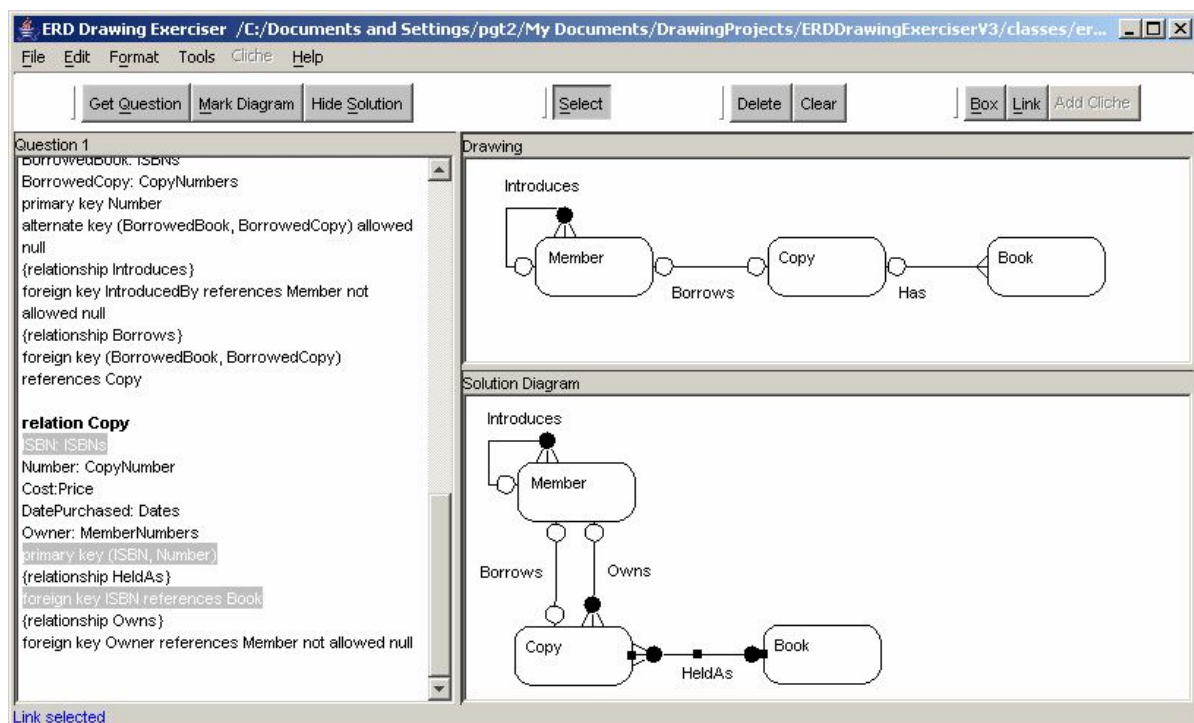


Figure 6 The revision tool

The bottom diagram, normally blanked-out, is the specimen solution whose features are all selectable. When the user selects one (the relation 'HeldAs' is shown selected), those parts of the text that relate to the feature are highlighted. The user can also right-click on a feature to obtain an explanation of how that feature relates to the problem.

When the user submits a diagram for marking, three elements of feedback are provided:

- 1 A grade that indicates how well the student's attempt matches the specimen solution;
- 2 A list of MMUs in the two diagrams showing how the automatic marker has matched those in the student diagram with those in the specimen solution;
- 3 A copy of the student's answer overlaying the specimen solution showing the similarities and differences between the two diagrams.

The revision tool has been tested with students (ref) and found to be a very welcome addition to our database course.

Authoring tool

The revision tool requires a considerable amount of data to be generated by the instructor including:

- 1 The questions (we currently provide 10);
- 2 Specimen solutions;
- 3 Descriptions of all the MMUs in each specimen solution;
- 4 The correspondences between the question text and the specimen solution.

To facilitate the production of this data we have built an authoring tool (see Figure 7).

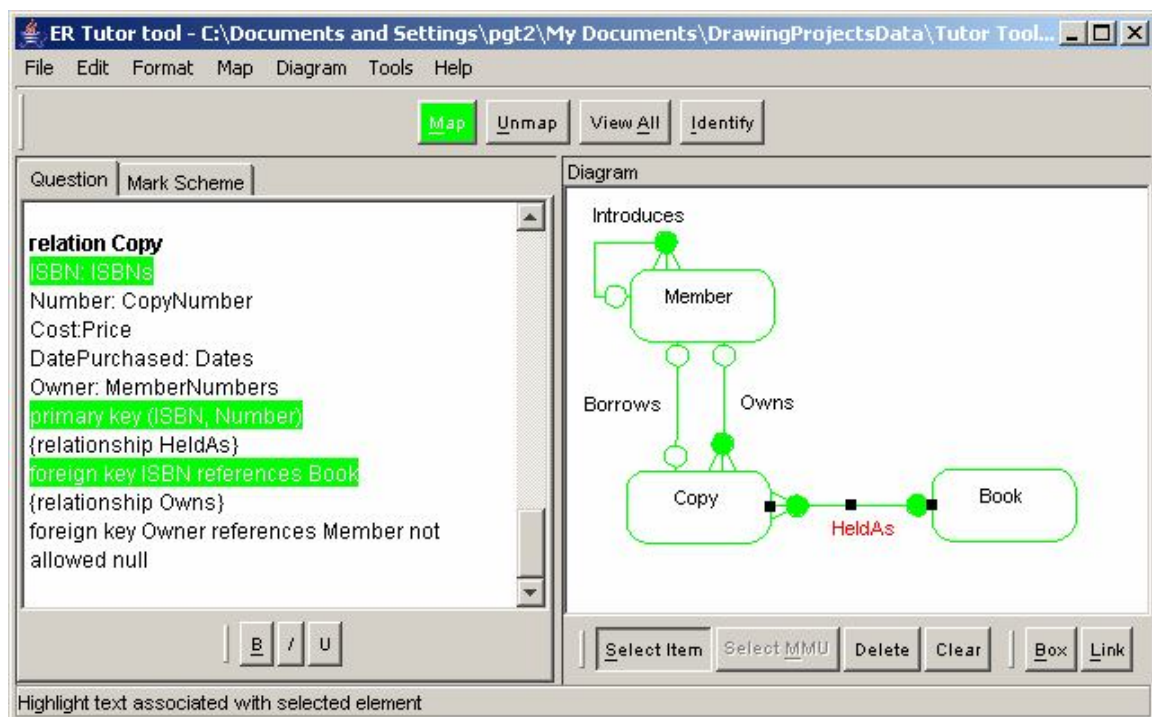


Figure 7 The authoring tool

The authoring tool has two main windows: one (on the left) for entering and editing the question text, the other (on the right) for entering and editing the specimen solution. The author can select a diagram feature (such as the 'HeldAs' relationship), highlight a section of

the question text and create an association (mapping) between them (using the ‘Map’ button). Right-clicking on a diagram feature reveals a template for describing the feature.

The authoring tool has recently been extended to provide a mechanism for specifying mark schemes (see Figure 8).

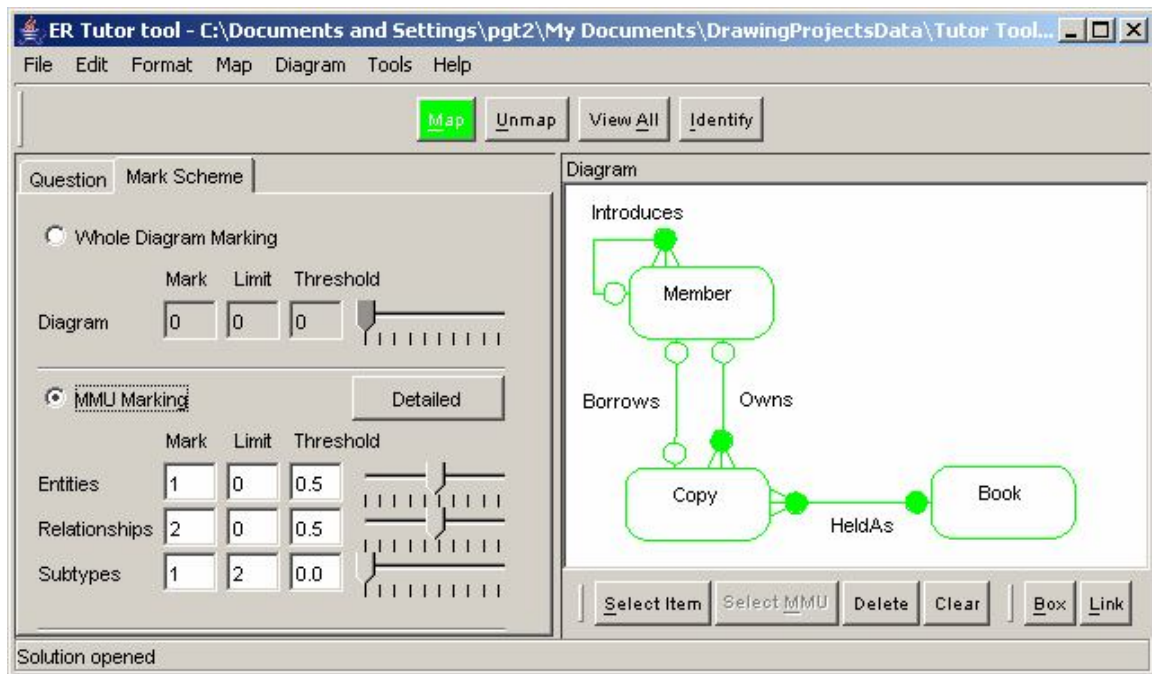


Figure 8 Specifying a mark scheme

In Figure 8, an example of ‘MMU marking’ is shown in which the user can specify a mark for each correct occurrence of each type of MMU. The limit parameter sets a maximum mark for all occurrences of an MMU type (to cope with marking schemes that state, for example, ‘award 1 mark for each correct instance of a subtype up to a maximum of 2 marks’). The threshold is a value in the range [0..1] which specifies how well an MMU in the student answer must match an MMU in the specimen solution to be considered worthy of receiving credit (an entity in the student answer will only receive a proportion of the single mark on offer if it is found to match with a solution entity type with a similarity greater than or equal to 0.5). The proportion of the marks on offer actually awarded to an MMU is determined by the similarity measure and, if requested, rounded to the nearest half-mark.

This scheme treats all instances of an MMU type equally. In mark schemes where specific MMUs are to be weighted differently, the tool allows marks to be assigned to MMUs individually – the ‘Detailed’ button reveals a template for doing this.

Assessment tools

The automatic marker is available as a stand-alone tool primarily aimed at batch processing. In this form, it is capable of providing the teacher with useful feedback on the performance of students. For example, it can show the frequency with which MMUs were found to be missing from student answers. Table 4 shows the frequency of missing relationships in the development set from our large corpus.

It is clear from this data that a large proportion of students had difficulty recognizing the relationships named ‘Has’ and ‘Is’ which might indicate some misunderstanding that needs to be followed up in the teaching.

Table 4 Missing relationships

Relationship	Count	%
Receives(Client, Presentation)	26	13.5
Gives(Employee, Presentation)	18	9.3
Prepares(Employee, Unit)	5	2.6
BelongsTo(Unit, Course)	8	4.1
Has(Course, Prerequisite)	140	72.5
Is(Course, Prerequisite)	158	81.9
Presents(Presentation, Course)	77	39.9

During marking, multiple alternative solutions (not all of which need to be entirely accurate, but which deserve some credit) are often discovered. Depending at what stage the marking has reached and the size of the cohort, such discoveries can be a significant overhead in the marking process. With an automatic marker it is a straightforward and speedy matter to draw the alternative solution and associate a mark scheme with it (using the authoring tool, for example) and set the marker to compare each answer against all solutions and choose the appropriate mark. Our automatic marker assumes that, in general, there will be multiple solutions to a question.

At its current stage of development, one might feel that the technology is not sufficiently reliable to be used alone in a summative environment. We envisage, however, that it could be used as a 'second marker' and its results compared to those of a human marker. In cases where there is a significant discrepancy between the human and the automatic marker, a second human opinion can be requested and hence reduce the overhead in 'double marking' by humans. Thus, the automatic marker can be viewed as a marking assistant.

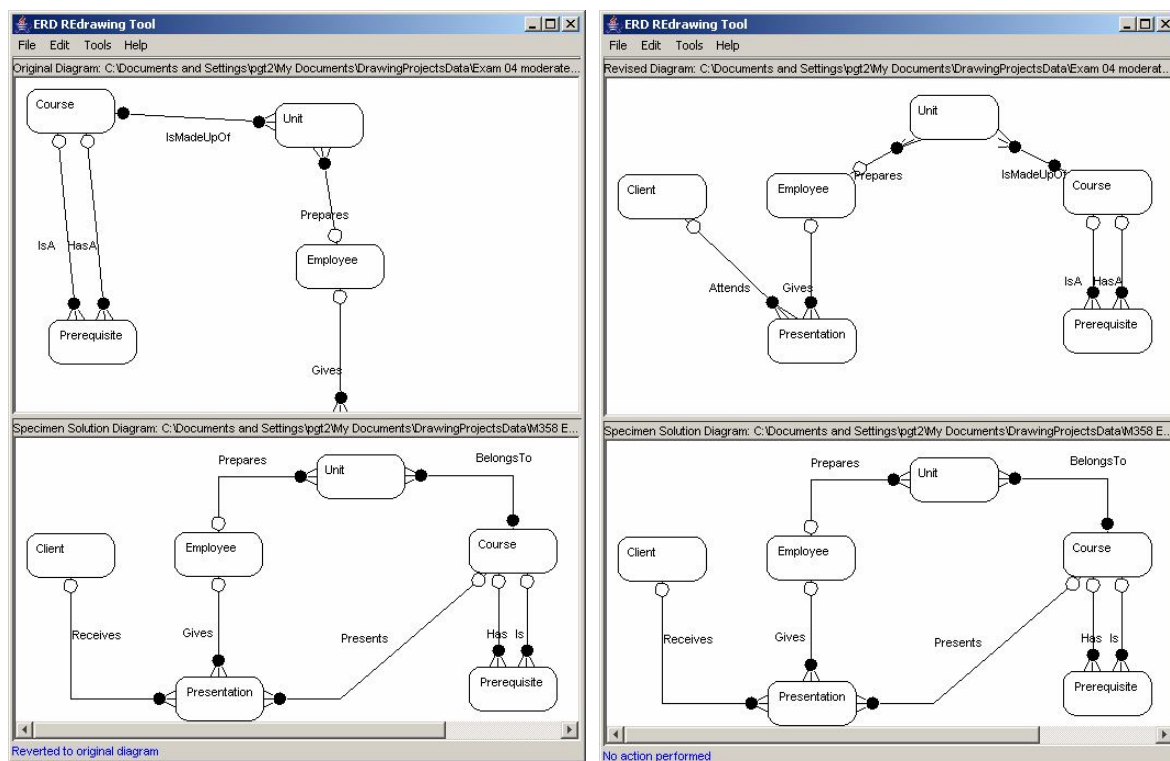


Figure 9 The marker's redrawing assistant

Pursuing the notion that the technology could be used to assist a human marker, we considered those cases where there was a discrepancy between the human and automated marking. In many of these cases it was difficult to decipher the student diagram because its structure was quite different to that of the specimen solution. We felt that the layout of the student answer was making it difficult for the human marker to recognize where marks should be awarded. There is evidence that even experts in the field can misinterpret diagrams (Hungerford et al., 2004). Therefore, we have begun to develop a ‘marker’s assistant’ that redraws a student’s answer diagram in a form that more closely resembles the specimen solution and hence makes the diagram easier to assess. Figure 9 shows two screen shots of the interface to the prototype assistant.

The redrawing tool, Figure 9, shows the student’s answer in the top pane and the specimen solution in the bottom pane. In the left hand example, the student’s answer is shown in its original form and in the right hand example the answer has been automatically redrawn to more closely resemble the specimen solution. It is easier to see the correspondences in the right hand example. The redrawing tool is an extension of the feedback mechanism built into the revision tool discussed earlier.

Conclusion and future work

The experiments we have performed to-date on the automatic marking tool have been encouraging. We have a framework for capturing, processing and interpreting graph-based diagrams that works in a practical learning situation. The interpretation phase applied to the marking of ERDs and compared with human markers has been effective, but we would like to improve some aspects of its performance by increasing the use of NLP algorithms (Manning & Schutze, 2002).

There are a small number of example diagrams in our corpora that are not dealt with sufficiently well by the automatic marking algorithm. In some cases, the deficiency is due to the lack of a more global view of the diagrams. In effect, we focus on MMUs not the larger MUs that might be present in the diagrams. Since we have incorporated very little of the aggregation stage of the framework into the marking process it is likely that we will overlook aggregations that may plausibly be equivalent to MUs in the specimen solution. Incorporating knowledge about aggregate equivalences and the way in which one instance of an MMU could contribute useful information when attempting to match other instances of MMU should improve the performance. A start has been made on this aspect and is reported in (Thomas et al., 2006).

We have also begun to investigate the use of the technology in other domains. The revision tool has been modified in a minor way to allow it to be configured with different diagramming notations while retaining its fundamental graph-based properties. Figure 10 shows its use for revising biological flows. In this domain, the weights must be set to reflect the relative importance of the flows to the biological entities.

Acknowledgement

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.

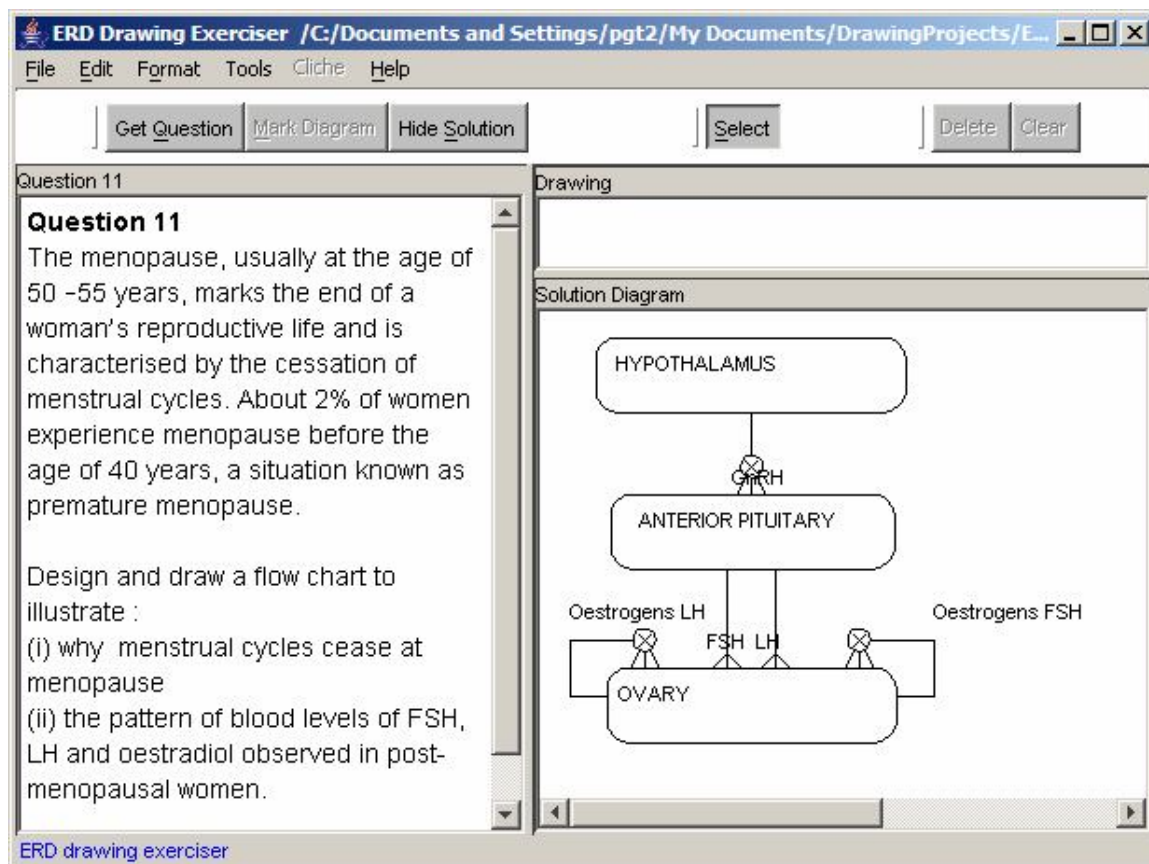


Figure 10 The use of the revision tool in the biological flow domain

References

- Anderson, M., McCartney, R. (2003) Diagram processing: Computing with Diagrams. *Artificial Intelligence* **145** (1-2) 181-226.
- Batmaz, F. and Hinde, C.J. (2006) A Diagram Drawing Tool for Semi-automatic assessment of Conceptual Database Diagrams. *Proceedings of the 10th Annual International Conference in Computer Assisted Assessment*. (Loughborough University, Loughborough, UK, July 2006), 68-81.
- Burstein, J., Chodorow, M. and Leacock, C. (2003) Criterion SM Online Essay Evaluation: An Application for Automated Evaluation of Student Essays. *Proceedings of the Fifteenth Annual Conference on Innovative Applications of Artificial Intelligence*, (Acapulco, Mexico. August 2003).
- Chok, S.S. and Marriott, K. (1995) Parsing visual languages. *Proceedings of the Eighteenth Australian Computer Science Conference*, Australian Computer Science Communications, **17**, 90-98.
- Haley, Debra Trusso, Pete Thomas, Anne De Roeck, and 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, 2005), 575-579.
- Higgins, C. A. and Bligh. B. (2006) Formative Computer Based Assessment in Diagram Based Domains. *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE)*, Bologna, Italy, June 26-28, 2006), 98-102.

- Hungerford B. C., Hevner A.R. and Collins R.W (2004) Reviewing Software Diagrams: A Cognitive Study, *IEEE Transactions on Software Engineering* **30** (2) Feb 2004 82-96.
- Kniveton, B. H. A. (1996) A correlational analysis of multiple-choice and essay assessment measures. *Research in Education*, **56**, 73-84.
- Manning, C.D and Schutze, H. (2002) *Foundations of Statistical Natural language Processing*. MIT Press, Cambridge, Massachusetts, USA. ISBN 0-262-13360-1.
- Marriott, K., Meyer, B. and Wittenburg, K.B. (1998) A survey of Visual Language Specification and Recognition. In *Visual Language Theory*, eds: Marriott, K and Meyer, B., Springer-Verlag, New York, 8-85, ISBN 0-378-98367-8.
- Newstead, S.E. and Dennis, I. (1994) The reliability of exam marking in psychology: examiners examined. *Psychologist*, **7** (5) 216-219.
- 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. and Waugh, K. (2004) Interpreting Imprecise Diagrams. *Proceedings of the Third International Conference in the Theory and Application of Diagrams*. March 22-24, Cambridge, UK. Springer Lecture Notes in Computer Science, eds: Alan Blackwell, Kim Marriott, Atsushi Shimojima, 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, *Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE, Thessaloniki, Greece, 2003)*, 50-54
- Thomas, P.G. (2004) Drawing Diagrams in an Online Exam, *Proceedings of the 8th Annual International Conference in Computer Assisted Assessment*. (Loughborough University, Loughborough, UK, 2004), 403-413.
- Thomas, P.G., Waugh, K., Smith, N. (2005) Experiments in the Automatic marking of E-R Diagrams. *Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE, Monte de Caparica, Portugal, 2005)*, 158-162.
- Thomas, P.G., Waugh, K., Smith, N. (2006) Using Patterns in the Automatic Marking of ER-Diagrams. *Proceedings of the 11th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE, June 26-28, Bologna, Italy, 2006)*, 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. *Proceedings of the 2nd LTSN-ICS Teaching, Learning and Assessment in Databases Workshop, Edinburgh*.

<http://www.ics.heacademy.ac.uk/events/displayevent.php?id=68> (accessed 08/02/07)