Using Patterns in the Automatic Marking of ER-Diagrams

Pete Thomas
Department of Computing
Open University
Milton Keynes, UK
+44 (0)1908652695
p.g.thomas@open.ac.uk

Kevin Waugh
Department of Computing
Open University
Milton Keynes, UK
+44 (0)1908653187
k.waugh@open.ac.uk

Neil Smith
Department of Computing
Open University
Milton Keynes, UK
+44 (0)1908654101
n.smith@open.ac.uk

ABSTRACT
This paper illustrates how the notion of pattern can be used in the automatic analysis and synthesis of diagrams, applied particularly to the automatic marking of ER-diagrams. The paper describes how diagram patterns fit into a general framework for diagram interpretation and provides examples of how patterns can be exploited in other fields. Diagram patterns are defined and specified within the area of ER-diagrams. The paper also shows how patterns are being exploited in a revision tool for understanding ER-diagrams.

Categories and Subject Descriptors
K.3.2 [**Computer and Information Systems Education**]: computer science education.

General Terms
Experimentation.

Keywords
Diagram interpretation, automatic grading, entity-relationship diagrams, patterns, teaching tool.

1. INTRODUCTION
In a previous paper [15] we presented an approach to the computer interpretation of diagrams and showed how it could be successfully applied to the automatic marking (grading) of student attempts at drawing entity-relationship (ER) diagrams. The automatic marker was incorporated into a revision tool to enable students to practice ER diagramming in the context of data models for database and obtain feedback on their attempts. In that paper we discussed the idea of *imprecise diagrams* in which the required features are either malformed (in the sense that it does not conform to standard rules for drawing specific features in the given domain) or missing, or extraneous features are included. Imprecise diagrams frequently occur in student answers to assignment questions.

ITiCSE'06, June 26–28, 2006, Bologna, Italy.

Our initial approach to marking diagrams was similar to the approach we adopted in the automatic grading of free-form text assignment answers [12, 13] in that we did not attempt to address any higher-order semantic structures. That is, the approach was equivalent to looking for key words and phrases in a sentential answer. The automatic grading of answers in textual form has received much attention over recent years [2, 3, 10].

More recently, we have developed an approach to the general problem of diagram understanding in the form of a five-stage framework that includes a stage which deals with higher-order constructs [11]. In this paper, we discuss an extension of the work in [15] to show how patterns in diagrams can contribute to the interpretation and automatic marking of E-R diagrams.

2. DIAGRAM INTERPRETATION
The approach we have taken to the interpretation of diagrams [11] is a framework of five stages which we have named segmentation, assimilation, identification, aggregation and integration. The first two stages translate a raster-based image into a set of diagrammatic primitives such as boxes, lines and text. In our teaching environment we use a diagramming tool which produces a representation of a diagram in terms of these primitives.

The identification stage identifies what we have called minimal meaningful units (MMUs). A box in an ER-diagram is an MMU because it represents an entity (an atomic element of the diagram). Two boxes connected by a line in an ER-diagram are an MMU because they represent a relationship between two entities. However, in another domain an association between two items denoted by a line joining two boxes might still be an MMU but its meaning would be different from that of a relationship in an ER-diagram. Thus, the meaning of a diagram is domain specific. The identification stage identifies all MMUs contained within the set of diagrammatic primitives using domain knowledge.

In this paper we shall concentrate on the aggregation stage in which MMUs are combined into higher level, abstract features. Once again, this stage is domain specific.

The final stage, interpretation, looks for meaning in a diagram. In our automatic marking application, meaning is ascribed to a student generated diagram through comparison with a specimen solution (another diagram represented as a set of abstract features) for which a grade, based on the degree of similarity, is generated.

While it would be possible to design a drawing tool to enforce the diagramming rules in a given domain (and certainly one might wish to do this in a teaching context), we decided that, for assessment purposes, there should be some latitude in what a tool

would accept because we are interested in discovering what the student knows, not how well the tool supports the construction of correct diagrams.

3. PATTERNS AND THEIR USES

In this work, a pattern can be viewed as a diagram with some of its details omitted (an abstract diagram), and is sometimes referred to as a cliché. To make this notion clearer, this section provides a number of motivating examples of the use of patterns in diagram manipulation.

3.1 Equivalent diagrams

In ER-diagrams, there are occasions when two diagrams with different structures can be considered to be equivalent. For example, in a data model for a database, a many-to-many relationship is often replaced by two one-to-many relationships (by introducing a new entity) as illustrated in Figure 1.

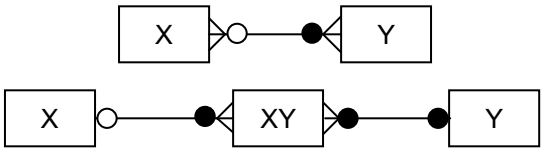

Figure 1. Equivalent diagrams.

Such equivalences are not necessarily reflexive: there are constraints on when such equivalences can be applied automatically. In the example shown in Figure 1, the conversion from a pair of one-to-many relationships to a single many-to-many relationship would only be allowed (in a data model) if the primary key (x, y) of the common entity XY consists of the primary keys of X (x) and Y (y) alone.

Several such equivalences exist, but again they cannot be applied in all circumstances. Figure 2 shows another pair of (sometimes) equivalent diagrams which represents the notion that if there are many courses in a programme of study and each course can have many set books, there must be many set books used in a programme.

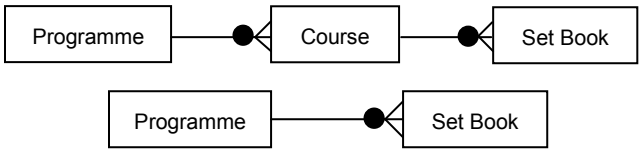

Figure 2. Equivalent diagrams.

These examples suggest that it would be worth searching in diagrams for parts that match patterns such as 'many-to-many relationships' and 'one-to-many followed by one-to-many relationships' and being able to replace one pattern occurrence by another pattern occurrence.

3.2 Automatic grading

In terms of grading, the ability to identify sub-diagrams within a diagram enables a number of possibilities. In automatic marking, one can envisage marking schemes that award differential marks to sub-diagrams. In manual marking, a tool that can identify sub-diagrams can be helpful to the marker (we have some evidence that suggests that humans can have difficulty in identifying sub-diagrams, particularly when the 'shape' of a student-produced diagram is quite different from the solution diagram).

3.3 Design patterns

From our perspective, UML class diagrams are quite similar to ER-diagrams – they have more structure, but are essential graph-based – and can be treated alike. Design patterns such as those found in [6] are described in terms of class diagrams and therefore it should be possible to search software design diagrams to find design patterns which had not previously been recognized and hence, through refactoring, improve the design.

3.4 Diagram construction

When constructing diagrams, having a library of diagrams (clichés) from which new diagrams can be created has many advantages ranging from ease of use to improved accuracy.

3.5 Patterns, clichés and sub-diagrams

Thus, in our view, a diagram is a set of sub-diagrams and when useful re-usable diagrams are stored in a library we refer to them as clichés. However, greater flexibility is achieved when diagrams and clichés are described in terms of patterns. That is, a pattern describes the general shape of a diagram and allows the user (human or machine) to fill in details and hence specialize the diagram. More generally, a sub-diagram or cliché may match a given pattern.

When viewing a diagram as a set of relationships, there is no requirement that a diagram should be a connected graph. A diagram can consist of a set of disconnected (sub-) diagrams. Similarly, there is no reason to consider a pattern to be connected; it could consist of a number of disconnected patterns.

Hence, we can envisage two broad uses of patterns: (a) analysing a diagram into sub-diagrams, and (b) synthesizing a diagram from sub-diagrams and the general problem is to find all occurrences of a set of patterns within a set of diagrams. Clearly, such a search may find zero, one or more matches.

4. SPECIFYING PATTERNS

In the area of ER-diagrams, simple patterns consist of single relationships such as 'one-to-many', 'many-to-many' which can be represented diagrammatically as shown in Figure 3. These patterns do not involve names (of the entities or the relationships) nor do they specify any participation constraints.

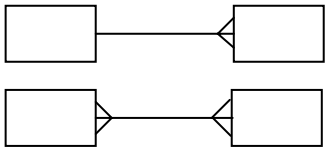

Figure 3. Simple patterns in ER-diagrams.

However, since names play an important role in distinguishing the two entities in a relationship, we allow patterns to have entity names which are distinguished from normal names by starting them with the underscore character (see Figure 4).

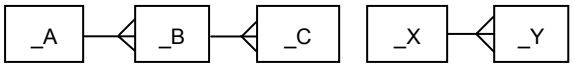

Figure 4. A pair of patterns.

Thus, if a task is to identify all occurrences of sub-structures in the diagram shown in Figure 5 that match the two patterns of Figure 4, the result is shown in Figure 6.

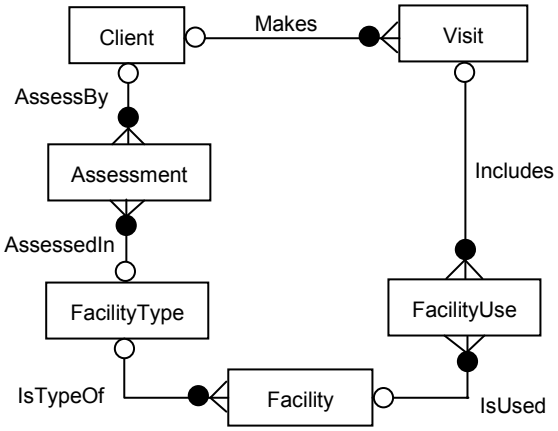

Figure 5. A sample ER-diagram.

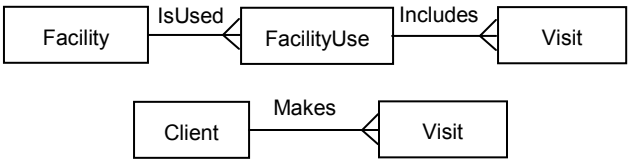

Figure 6. Matches for the patterns in Figure 4.

The result in Figure 6 can also be represented as:

IsUsed(Facility, FacilityUse), Includes(FacilityUse, Visit)

Makes(Client , Visit)

When dealing with patterns, the conventions for drawing an ER-diagram require a small extension. In an ER-diagram, a relationship consists of an association, denoted by a line, between two entities each denoted by a rectangle. A relationship can be 'adorned' at each end with a degree type (many or one, denoted by the presence or absence of a 'crowsfoot') and a participation type (selected from none, open and closed and denoted by a circle). However, since a pattern is a relationship in which some of the attributes are unspecified we need a way of denoting that a participation type is unspecified. Simply omitting a participation type conventionally means 'none'. We need a denotation for 'anything' and have adopted an open circle with a cross inside. We have assumed that the degree of a relationship in a pattern will always be specified in a pattern and that the notion of a pattern that matches any degree is not required.

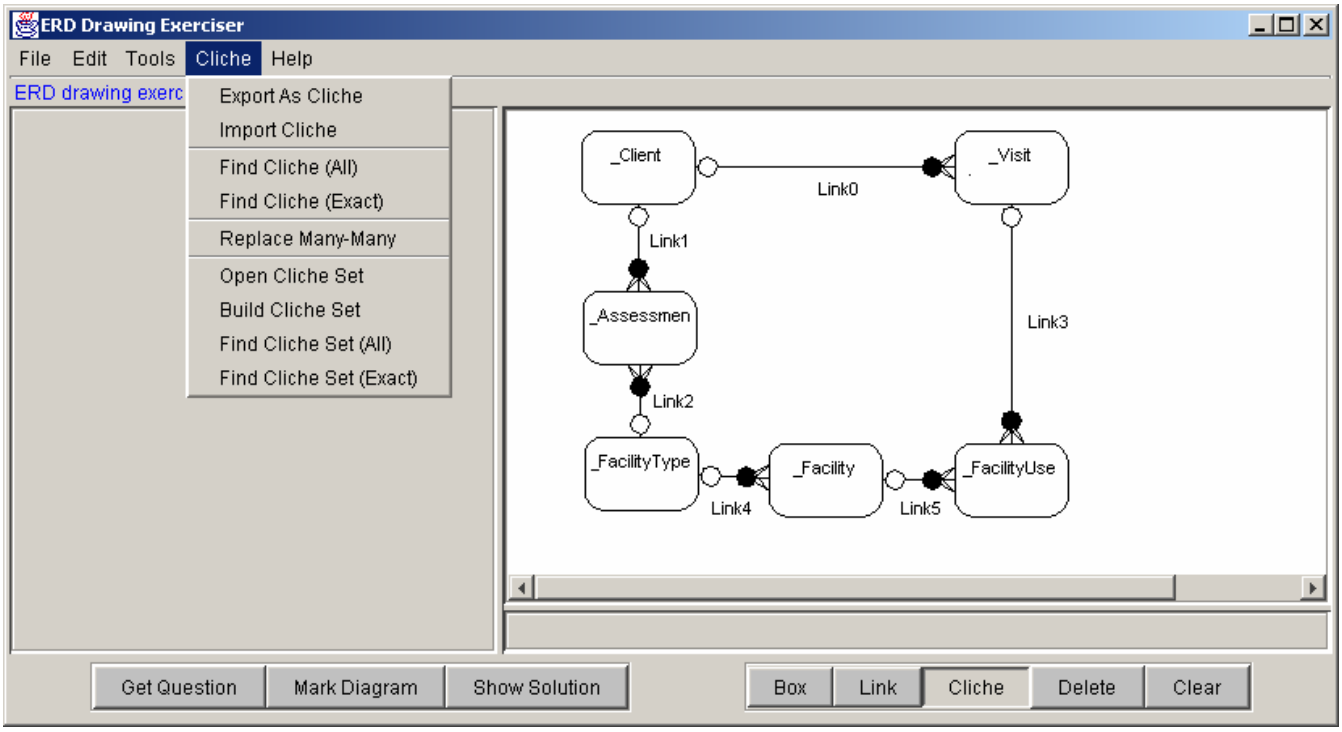

Figure 7. The Revision Tool.

5. THE TUTORIAL TOOL

One application of our automatic marker is a tutorial tool (see Figure 7). The tool displays a typical data modelling question in the left-hand pane and students are asked to draw the corresponding ER-diagram in the top-right-hand pane. Once the student has completed a diagram the tool can be asked to mark and comment on the accuracy of the diagram (the Mark Diagram button in Figure 7):

To investigate the ideas behind patterns, the tutorial tool has been extended with the following functionality (see Figure 7):

- create and store individual patterns (named clichés in the tool) in a library;
- add patterns (clichés) to a diagram from a library;
- find all occurrences of a specific pattern exactly – only exact matches for the pattern are returned;
- find all approximate matches to a pattern – the similarity measure between the pattern and returned sub-diagram is also returned;
- search for all many-to-many relationships in a diagram and suggests replacement by two one-to-many relationships;
- build a cliché set;
- find a match for a set of clichés.

6. EXACT AND INEXACT MATCHING

In the discussion so far we have assumed that a search for a pattern will result in a set of exact matches. That is, if one were searching for a one-to-many relationship, this would not match with a many-to-many relationship, but it would match with a many-to-one relationship. However, when working with imprecise diagrams (for grading purposes, say) it may be necessary to relax the exact criterion and be prepared to accept 'close' matches. Measures of closeness then need to be devised which would allow accepted matches to be ordered and the 'best' match to be determined.

Our current approach to matching individual relationships compares each relationship in one diagram with each relationship in another diagram to produce a similarity matrix. The similarity matrix is used to determine the 'best' match – the correspondence between the two sets of relationships that maximises a similarity measure – we currently use the sum of the similarities between pairs of relationships.

A fundamental change is required to this algorithm when patterns are present. It is possible for a cliché in one diagram to match exactly with more than one cliché in the other diagram, making the search space for the best match much bigger and possibly yielding more than one 'best' match. For example, a pattern consisting of a many-to-many cliché will match with a many-to-many relationship in two ways.

One of our research questions is to determine the most appropriate similarity measure. Currently we have adopted the following scheme.

Let c1 and c2 be two clichés. Then the similarity between them is given by:

sim(c1,c2) = w*sim(entites(c1), entities(c2)) +
(1-w)*sim(adornments(c1), adornments(c2))

The expression

sim(entites(c1), entities(c2))

is a measure of the similarity between the entities of c1 and the entities of c2. This reduces to comparing the similarity of the names used for the entities for which we use an algorithm based on stemming and edit distance. The expression

sim(adornments(c1), adornments(c2))

is a measure of the similarity between the adornments (the degrees and participations) of c1 and the adornments of c2. The parameter w is currently 0.6 (found by experiment).

The problem of finding matches for one-to-many relationships in the diagram given in Figure 8 gives the results shown in Figure 9.

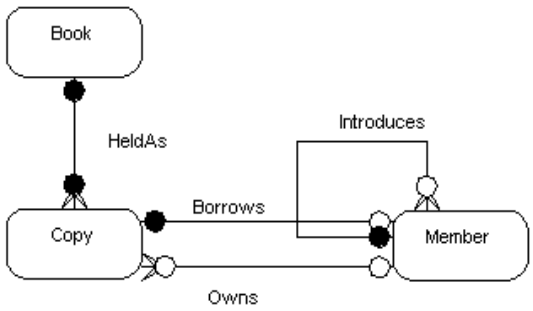

Figure 8. A sample ER-diagram.

0	HeldAs(Book, Copy)	1.0
1	HeldAs(Copy, Book)	0.75
2	Borrows(Copy, Member)	0.75
3	Borrows(Member, Copy)	0.75
4	Owns(Copy, Member)	0.75
5	Owns(Member, Copy)	1.0
6	Introduces(Member, Member)	1.0
7	Introduces(Member, Member)	0.75

Figure 9. Exact and inexact matches.

7. SUMMARY AND FUTURE WORK

The introduction of the notion of patterns has provided a mechanism for building up diagrams from the basic building blocks of MMUs – effectively creating new meaningful units (MUs). So far, we have exploited this capability by expanding the capability of a revision tool to create libraries of MUs that can be subsequently employed to construct new diagrams. The tool has also been extended with the capability of searching for sub-diagrams both exactly and inexactly (to give the 'best' match). This capability is an essential component of an automatic marker with higher-order semantic 'understanding'.

Our plan for the future has four strands. One is to extend the existing automatic marker with an aggregation stage and apply it to examples with richer structures than those studied to date. We would like to apply this technology to describe common mistakes made by students and hence to provide improved feedback.

The second strand is the construction of a large corpus of student answers to diagramming problems on which to test the marker. To-date we have a corpus of around 600 examples which will shortly be expanded to over a thousand and hence support our in-depth testing and development of the marker. The third strand is to apply this work to more complex domains such as UML diagrams. This latter work will lead into the fourth strand - looking for design patterns in software design diagrams.

8. 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.

9. REFERENCES

[1] Anderson, M., McCartney, R. (2003) *Diagram processing: Computing with Diagrams*. Artificial Intelligence **145** (1-2) 181-226.

[2] Burstein, J., C. Leacock, et al. (2001) Automated Evaluation of Essays and Short Answers. In *Proceedings Fifth International Computer Assisted Assessment Conference*, Loughborough University, UK, Learning & Teaching Development, Loughborough University, 41-45.

[3] Burstein, J., Chodorow, M. and 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.

[4] Chok, S.S. and Marriott, K. (1995) Parsing visual languages. In *Proceedings of the Eighteenth Australian Computer Science Conference*, Australian Computer Science Communications, **17**, 90-98.

[5] Donlon, J.J., Forbus, K.D. (1999) Using a geographic information system for qualitative spatial reasoning about traceability. In *Proceedings of the Qualitative Reasoning Workshop*, Loch Awe, Scotland.

[6] Gamma, E., Helm, R., Johnson, R and Vlissides, J. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison Wesley. ISBN 0-201-63361-2.

[7] Iizuka, K., Tanaka, J. and 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)

[8] Jamnik, M. (1998) Automatic Diagrammatic proofs of Arithmetic Arguments. PhD Thesis, University of Edinburgh.

[9] 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.

[10] 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.

[11] Smith, N, Thomas, P.G. and 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. Springer Lecture Notes in Computer Science, eds: Alan Blackwell, Kim Marriott, Atsushi Shimojima, 2980**,** 239-241. ISBN 3-540-21268-X.

[12] 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.

[13] 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), Thesaloniki, Greece, 50-54.

[14] 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.

[15] 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.

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

[17] 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. http://www.ics.ltsn.ac.uk/pub/databases04/index.html (accessed 15/01/06)