THE ORM MODEL AS A KNOWLEDGE REPRESENTATION FOR E-TUTORIAL SYSTEMS

Thanaporn Leelawatananon
Faculty of Information Technology,
King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520, Thailand,

Suphamit Chittayasothorn
Faculty of Engineering,
King Mongkut's Institute of Technology, Ladkrabang, Bangkok 10520, Thailand,

Keywords: Object Role Model, Knowledge-based System, Knowledge Representation, e-Tutorial.

Abstract: At present information technology plays important roles in teaching and learning activities. E-learning systems have the potential to reduce operating costs and train more people. Teachers and students do not have to be in the same place at the same time and the students have the opportunity to perform self-studies and self-evaluation using e-tutorial systems. E-learning systems could be considered expert systems in the sense that they provide expert advice in particular subjects of studies to students. The exploitation of knowledge base and knowledge representation techniques is therefore vital to the development of e-learning systems. This paper presents the development of a knowledge-based e-tutorial system that uses the Object Role Model (ORM) as its knowledge representation. The system provides Physics tutorials. It was implemented in Prolog and the knowledge base is on a relational database server.

1 INTRODUCTION

Nowadays, the most common underlying data management facilities of e-learning systems are still the conventional files and record structures. Most e-learning systems interact directly with files and tables without knowledge representations. Recently, the use of ontology and knowledge representation models in e-learning systems appears in literatures. Resource Definition Framework (RDF)) (Brickley and Guha, 2000) and Ontology Interchange Language (OIL) (Studer, 2000) are used to represent knowledge in e-learning systems based on the Semantic Web (Stojanovic et al., 2001).

This paper presents an e-tutorial system which uses the Object Role Model (ORM) (Halpin, 2003) as a knowledge representation. The ORM model is a well-established conceptual schema model originated in Europe during the 1970s and was originally called NIAM (Verheijen and van Bekkum, 1982). It is a popular conceptual schema model for relational database design. The transformation from an ORM conceptual schema diagram to relational database schemas guarantees the project/join normal form (5NF).

Conventional knowledge-based systems that implement the knowledge base on a relational database use relational database schemas as predicate structures structures (Norrie et al., 1995; Albernethy and Altman, 1998; Maier et al., 2002). A tuple is perceived as a predicate instance. The problems of this approach are that a tuple of a relation may contain several facts only one of which is relevant and a tuple may have null values. Using the ORM conceptual schema as a knowledge representation solves these problems. An ORM fact type is a predicate structure and a fact instance becomes a predicate instance. Fact instances always have the truth value "true" following the 2-value logic and null value is not permitted in ORM. These properties make ORM a suitable knowledge representation for knowledge base systems implemented on large shared relational databases.

Leelawatananon T. and Chittayasothorn S. (2004).
THE ORM MODEL AS A KNOWLEDGE REPRESENTATION FOR E-TUTORIAL SYSTEMS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 479-484
DOI: 10.5220/0002655504790484

2 OBJECT ROLE MODEL

ORM is a fact-based conceptual schema model. Its main constructs are entity types, label types, fact types, reference types, subtype hierarchies and some static integrity constraints. It demonstrates clear distinction between the concept of entity types and the naming of the entity types (label types).

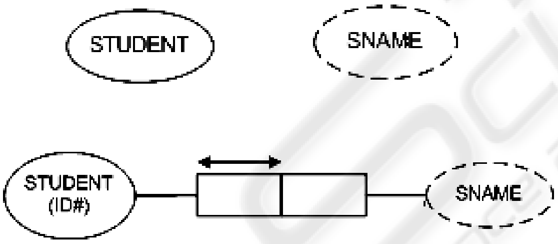
An entity type is an object of interest. An entity type may have several label types (value types) associate with it via reference types. Label types are naming of entity types and there is no concept of attributes in ORM. A one-to-one reference type is chosen to be the unique identifier of the entity type. Other reference types remain alternative identifiers of the entity type. Figure 1a shows an entity type STUDENT together with its label types SNAME and ID#. Figure 1b shows that ID# is chosen to be the unique identifier of STUDENT. The unique identifier is shown in brackets inside of the entity type. SNAME remains in a reference type. Relationships between entity types are called fact types. These fact types must be elementary (cannot be further decomposed). N-ary fact types are also allowed. A uniqueness constraint enforces uniqueness of entity instances participated in the fact type. A transformation algorithm from an ORM

allowed. A uniqueness constraint enforces uniqueness of entity instances participated in the fact type. A transformation algorithm from an ORM

conceptual schema to relational database schemas takes advantage of this non-decomposable property. Each fact type is perceived as a relational database schema in 5NF. Fact types associated with a common entity type and have the uniqueness constraint enforced on the entity type form a relational schema in 5NF (Halpin, 2003). Figure 1c shows ORM fact types and corresponding relational database schemas.

3 ORM AS A KNOWLEDGE REPRESENTATION MODEL

From the previous section, the ORM model is perceived as a relational database design tool that guarantees the 5NF relational database schemas. In this project, we use it as the logical structure that our e-learning system works on. ORM is the knowledge representation of our system.



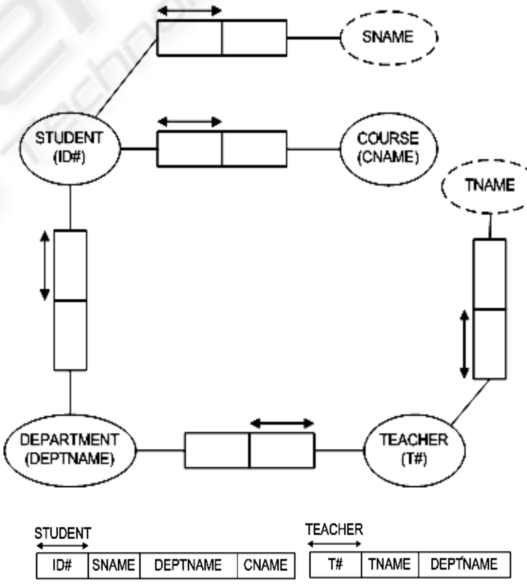


Figure 1: a) An entity type and label types. b) An entity type together with its unique identifier and a reference type. c) An ORM conceptual schema diagram and corresponding relation schemas.

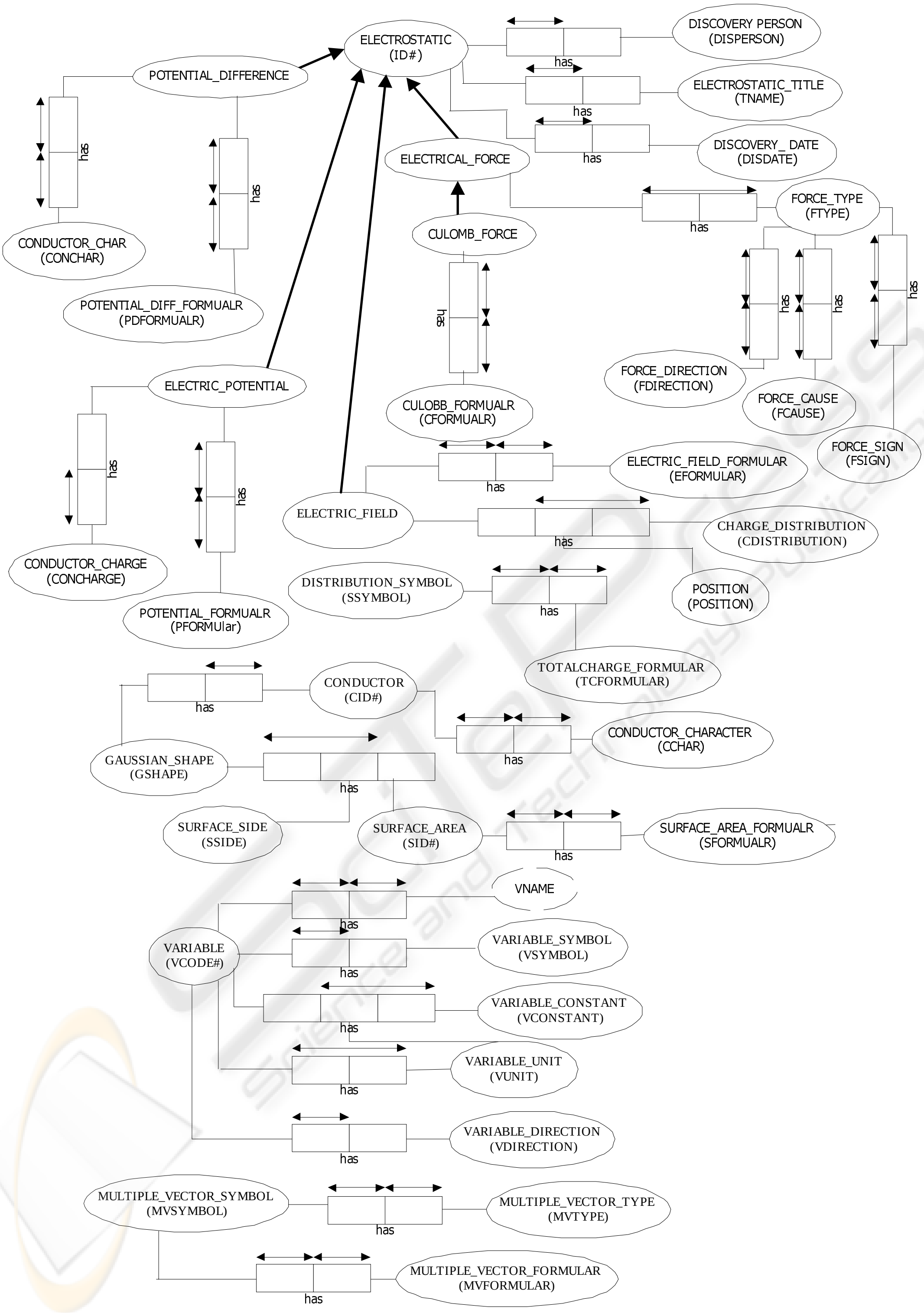


Figure 2: ORM diagram for the Gaussian electric field calculation

This approach has the advantage over the use of conventional knowledge representations such as Frames and Semantic Networks (Waterman, 1999) when implemented in relational database environments. There are no transformation algorithms that guarantee minimum redundancies and the 5NF (project/join normal form) from Frames and Semantic Networks. This is due to the fact that they are pointer-based representations. Most artificial intelligence systems implement them as Lisp programs. ORM, on the other hand, was developed for database modeling. The implementation of an ORM knowledge base using commercially available relational DBMS enables the knowledge base to be shared by many user applications. such as recovery control, concurrency control, indexing and query optimizations are readily available for our knowledge-based e-tutorial system. Figure 2 shows the ORM diagram for the Gaussian electric field calculation topics of the e-tutorial system. Another important point of using the ORM model as a knowledge representation is that each predicate instance corresponds with an ORM fact instance, not the entire tuple of a relation. This is Figure 2: ORM diagram for the Gaussian electric field calculation very useful since each tuple could contain many facts. Classical and more recent interfaces between expert systems and database systems refer to a relation as a predicate instance (Wang, 2000; Nick et al., 2001). This is not realistic in practice because there could be irrelevant facts on each tuple. It is proposed that a predicate instance refers to a fact instance of an ORM fact type.

4 ORM META CONCEPTUAL SCHEMA

An ORM meta conceptual schema is a conceptual schema that describes the ORM conceptual schema model. Since the users' ORM schema for domain knowledge must be stored on the database, a set of system tables is required to keep the information about the users' conceptual schemas. The meta conceptual schema is transformed into relational schemas for the system tables that keep information about the users' ORM schemas. Figure 3 shows the ORM meta conceptual schema which is used by our e-tutorial system.

5 AN E-TUTORIAL SYSTEM FOR PHYSICS

The prototype e-tutorial system presented in this paper gives Physics tutorials. It assists student's work on Physics exercise questions and evaluates students' understanding of the topic. The exercises are grouped in chapters. For each exercise, the system asks questions to guide the student to the solution of the problem. The questions are sequenced in the following order: questions on the formulae used for the given exercise, questions about relevant variables, questions about the main knowledge of the exercise and the application of formulae to obtain the result.

During a working session the system analyses the answer to each question to evaluate the student's understanding of the topic and shows the marks and evaluation result to the student. The system interacts with its ORM knowledge base to obtain related knowledge for the guidance and evaluations.

The system is implemented in WinProlog (Steel, 2000) and the underling DBMS is MS SQL*Server (Rebecca, 2000). ORM fact types are implemented as views and Prolog retrieves the content of the views when it consults the database. This means that the Prolog program is not aware of the underlying tables and refers to fact type views only. Each Prolog predicate instance an ORM fact instance, not the entire tuple of a relation. Figure 4 shows data flow diagrams that describe the e-tutorial system. Figure 5 shows a sample study session in Gaussian electric field calculation. The feature of our prototype e-tutorial system is comparable to other systems such as ANDES system (Joel, 2000), and another web-based tutorial system for engineering, mathematics and science subjects (Scott and Stone, 2000). Our prototype system guides the students step by step while some of these systems such as ANDES give the full guidance first and then let the students solve the problem afterwards. However, the features of the e tutorial system is not the main issue here. It is the ability to store Physics knowledge on relational database using the ORM model as the knowledge representation.

6 CONCLUSION

The e-tutorial system presented in this paper uses the ORM conceptual schema model as its knowledge representation. The system refers to fact instances of fact types when it analyzes student's answers and evaluates the level of understanding of the topic.

This is a new approach in knowledge-based system design and implementation on large shared databases. Developers use SQL to access the knowledge base. So, this approach is easy to

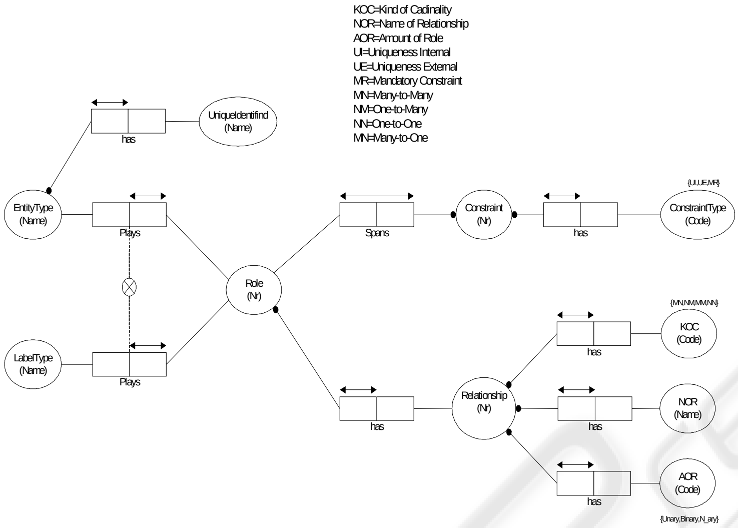


Figure 3: An ORM Meta Conceptual Schema

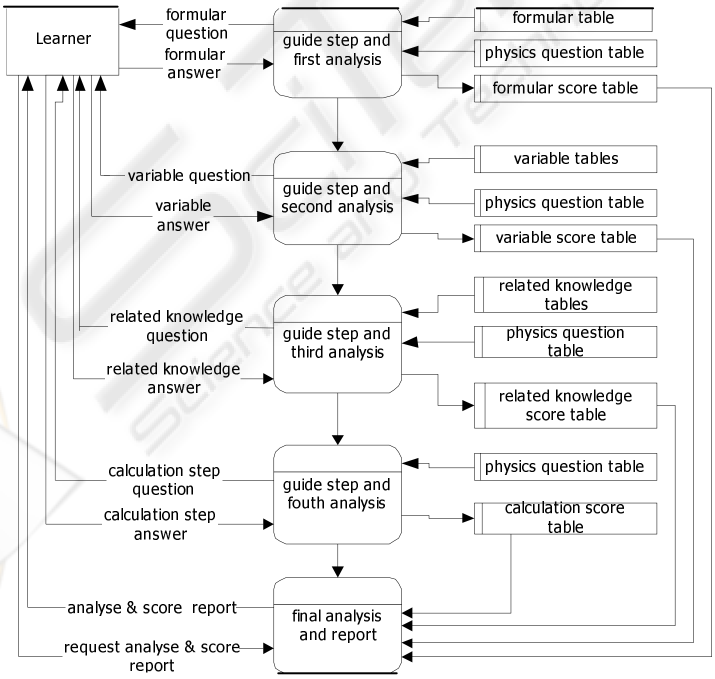


Figure 4: A data flow diagram for the e-tutorial system

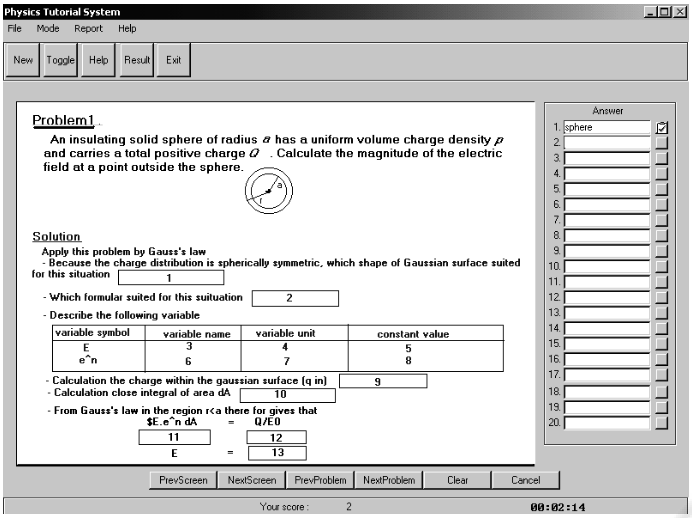


Figure 5: A sample study session in Gaussian electric field calculation

implement from the developer's point of view. The knowledge can be shared, reuse, and manage in multi-user environment by database servers.

REFERENCES

Abernethy, N., Altman, R., 1998. Sophia : Providing Basic Knowledge Service with A Common DBMS, Proceedings of the 5th KRDB Workshop.

Brickley, D. and Guha, R., 2000. Resource Description Framework (RDF) Scheme Specification 1.0. Candidate recommendation: World Wide Web Consortium. See http://www.w3.org/TR/2000/CR-rdf-scheme-20000327.

Halpin, T.A., 2003. Information Modeling and Relational Databases Using ORM with ER and UML: Morgan Kaufmann.

Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., van Harmelen, F., Klein, M., Staab, S., and Maier, F., Nute, D., Potter, W., Wang, J., Twery, M., Rauscher, M., Knopp, P., Thomasma, S., Dass, M. and Uchiyama, H., 2002. PROLOG/RDBMS Integration in the NED Intelligent Information System. In the proceedings of the Tenth International Conference on Cooperative Information Systems, Irvine, California.

Joel, A., 2000. A Computer Tutorial System for Introductory Physics Courses, See: http://www.physics.rutgers.edu/~shapiro/RutgK15/RutgK15.pdf

Nick B., Ioannis V., Ahmed K. and Elias N., 2001, InterBase: Integrating a knowledge Base System with a Multidatabase System for Data Warehousing, See: http://lpis.csd.auth.gr/publications/interbase-kb.pdf

Norrie, M., Reimer, U., Lippuner, P., Rys, M., Schek, H., 1995. Frames, Objects and Relations : Three Semantic Levels for Knowledge Base Systems. Swiss Priority Programme in Computer Science No. 5003-34347.

Rebecca Riordan, 2000. Microsoft® SQL Server™ 2000 Programming Step by Step.

Scott, N., and Stone, B., 2000. A Flexible Web-based Tutorial system for Engineering, Maths and Science Subjects, Global J. of Engng. Educ., Vol.2,No.1, Australia.

Steel, D., 2000 , Win-PROLOG Win 32 Programming Guide, See : http://www.lpa.co.uk

Stojanovic, L., Staab, S., Studer R., 2001. eLearning based on the Semantic Web. In: Webwet 2001-World Conference on the WWW and Internet, October 23-27, Orlando, Florida,US.

Studer, R., 2000. OIL: The Ontology Inference Layer. Technical report: University of Manchester / Vrije Universiteit Amsterdam. See http://www.ontoknowledge.org/oil/.

Verheijen, G. M. A., and J. van Bekkum, 1982. NIAM: An Information Analysis Method. Information Systems Design Methodologies: A Comparative Model. Proc. IFIP WG 8.1 Working Conference: North-Holland.

Wang L. and Wang M., 2000, Relational knowledge Representation Model in Intelligent Tutor System, See: http://www.pku.edu.cn/academic/xb/2000/_00e512.html

Waterman, D. A., 1999. A Guide to Expert Systems: Addison-Wesley.