An Automated Knowledge Acquisition Tool Based on the Inferential Modeling Technique

Christine W. Chan and Hanh H. Nguyen
Department of Computer Science/Energy Informatics Laboratory, University of Regina
University of Regina, Regina, Saskatchewan, Canada, S4S 0A2
Tel: (1-306) 585-5225, Fax: (1-306) 585-4745
e-mail: Christine.Chan@uregina.ca, nhongh@cs.uregina.ca

Abstract: Knowledge acquisition is the process that extracts the required knowledge from available sources, such as experts, textbooks and databases, for incorporation into a knowledge-based system. Knowledge acquisition is described as the first step in building expert systems and a major bottleneck in the efficient development and application of effective knowledge based expert systems. One cause of the problem is that the process of human reasoning we need to understand for knowledge-based system development is not available for direct observation. Moreover, the expertise of interest is typically not reportable due to the compilation of knowledge which results from extensive practice in a domain of problem solving activity. This is also a problem of modeling knowledge, which has been described as not a problem of accessing and translating what is known, but the familiar scientific and engineering problem of formalizing models for the first time. And this formalization process is especially difficult for knowledge engineers who are often faced with the difficult task of creating a knowledge model of a domain unfamiliar to them.

In this paper, we propose an automated knowledge acquisition tool which is based on an implementation of the Inferential Modeling Technique. The Inferential Modeling Technique is derived from the Inferential Model which is a domain-independent categorization of knowledge types and inferences [Chan 1992]. The model can serve as a template of the types of knowledge in a knowledge model of any domain.

1. Introduction

Knowledge acquisition is the process that extracts the required knowledge from available sources, such as experts, textbooks and databases, for incorporation into a knowledge-based system. Knowledge acquisition is crucial for the development of knowledge-based expert systems and is described as the first step in building expert systems and a major bottleneck in the efficient development and application of effective knowledge based expert systems. One cause of the problem is that the process of human reasoning we need to understand for knowledge-based system development is not available for direct observation. Moreover, the expertise of interest is typically not reportable due to the compilation of knowledge which results from extensive practice in a domain of problem solving activity. This is also a problem of modeling knowledge, which has been described as not a problem of accessing and translating what is known, but the familiar scientific and engineering problem of formalizing models for the first time. And this formalization process is especially difficult for knowledge engineers who are often faced with the difficult task of creating a knowledge model of a domain unfamiliar to them.

In the Knowledge Acquisition research community, the emphasis is on knowledge level modeling, which tackles the knowledge acquisition and analysis of a domain by means of various modeling efforts such as the KADS methodology, the generic task approach, and the Inferential modeling technique. Within the framework of knowledge-level modeling, two major lines of research have developed. One refines the existing knowledge-level frameworks and emphasize their formalizations. For example, ML2 has been developed as a formal implemented language based on the KADS methodology [Flores-Mendez et al. 1992]. Another line of research aims at developing knowledge level models for a range of tasks and domains in order to uncover generic components, problem solving methods, and ontologies that enable reuse across applications. The objective of the effort is to ease the KA problem by providing domain-independent generic models which can guide knowledge engineers in the construction of knowledge models for a particular domain. Specifically, this effort at knowledge modeling can proceed along one of two axis: (1) problem solving methods, and (2) domain ontology. Briefly, a problem solving method can be seen as an abstract model which provides a means of identifying at each step, candidate actions in a sequence of actions that accomplish some task within a specific domain [McDermott 1988]; while an ontology defines the vocabulary of representational terms with agreed upon definitions in human and machine readable forms [Gruber 1992].

In this paper, we propose an automated knowledge acquisition tool which is based on an implementation of the Inferential Modeling Technique. The Inferential Modeling Technique is derived from the Inferential Model which is a domain-independent categorization of knowledge types and inferences [Chan 1992]. The model can serve as a template of the types of knowledge in a knowledge model of a domain. The Inferential Modeling Techniques has been applied for analysis of knowledge in diverse domains including a legal domain [Chan 1995], a reforestation domain [Chan and Johnston 1996], a process design domain and a *pipeline monitoring* domain [Chan 2000].

The objective of developing the automated knowledge acquisition tool is to provide automated support to the knowledge acquisition process as a step towards construction of an ontology for a domain. An ontology is an explicit specification of a conceptualization. It provides a comprehensive foundation specification of knowledge in a domain. In the simplest case, an ontology can be represented as a hierarchy of concepts related by subsumption relations. In more complex cases, a variety of axioms can be added to express relationships and constraints among domain concepts.

2. Implementation

The Knowledge Acquisition (KA) System implemented based on the Inferential Modeling Technique has been designed to act as the front end system that supports acquiring and storing knowledge items on a domain. The repository of knowledge items constitutes a knowledge base, which can then be translated into portable representations such as XML, which can function as an ontology of knowledge-based systems.

The KA System consists of two main components: User Interface (UI) and Databases (DB). The UI receives input from users and stores it in the DB. It also retrieves data from the DB and displays it to the users. The system has been developed with JBuilder Professional 4[1], which provides facilities for easy development of user interface and database applications.

2.1 The User Interface

The UI allows users to create a new or open an existing ontology. To create a new ontology, users need to provide the ontology name, its authors, its domain and any associate documentation. The main screen of the UI is shown in Figure 1. To open an existing ontology, users can type its name in or choose from a file system dialog. The class-and-attribute window will be opened.

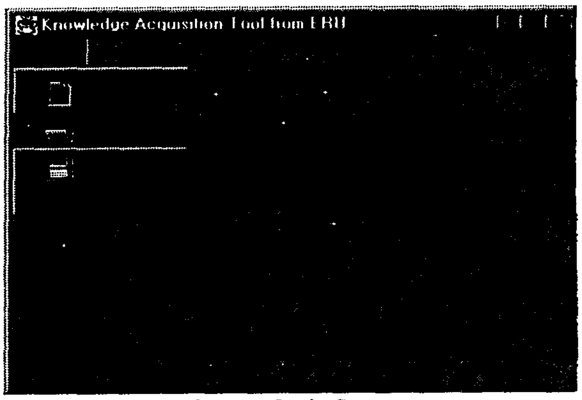

Figure 1. Main Screen

The class-and-attribute screen contains two tabs, one for class and one for attribute information. They are illustrated in Figures 2 and 3. Data retrieved from the database is reflected in the visual components in these screens. These visual components were implemented as dbSwing components. Most of these components are text fields with only a few exceptions. The relationship between super- and sub- classes is displayed with a Tree component (Figure 2) where subclasses are branches from a node of the tree. Class and attribute lists are implemented as List components.

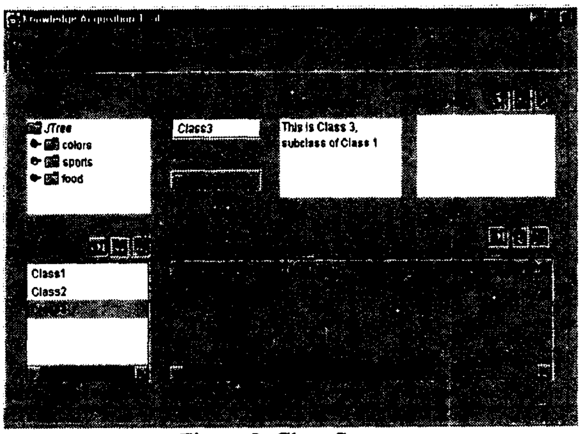

Figure 2. Class Screen

[1] Trademark of Borland USA

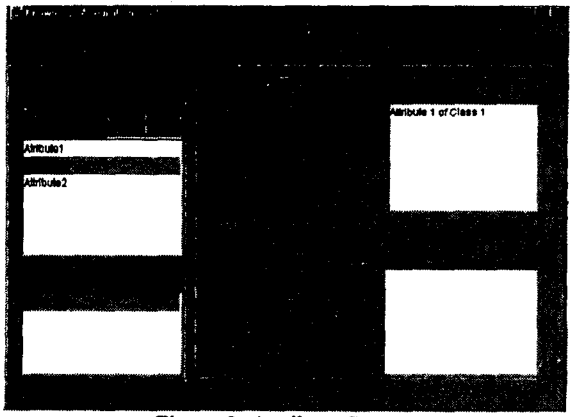

Figure 3. Attribute Screen

Visual components interact with each other. For example, when the user chooses an attribute from the attribute list in the attribute screen, the class list displays all the classes which contain the attribute. Other components that display the attribute's properties are also updated.

Each screen has a set of buttons that allow users to add, delete and modify the class and attribute properties. An editor has been constructed for editing the properties, a screen from the editor is shown in Figure 4.

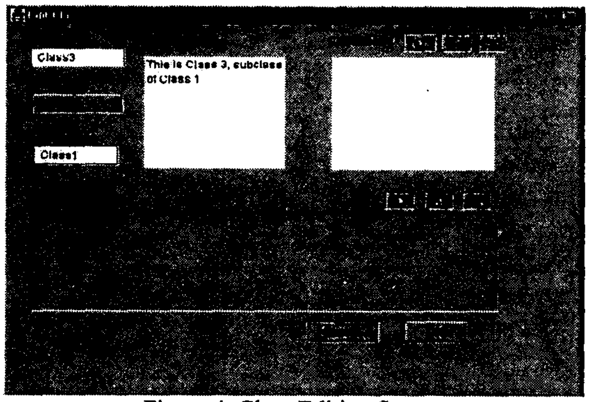

Figure 4. Class Editing Screen

2.2 The Database

The data received through the UI are stored in a storage facility called JDataStore. A JDataStore is a high performance, pure Java, portable, compact and persistent storage.

The database consists of three main tables: Ontology, Class and Attribute table.

- Ontology table stores information about an ontology. This includes ontology name, author name, domain of the ontology, documentation, creation date and root class from which all other classes are descended. The ontology name is the primary key.
- Class table contains class name, ontology name, the role of the class (abstract or concrete), its super class, any possible constraints and its documentation. Class name is the primary key.
- Attribute table stores attribute name, the name of the class to which it belongs, its value type, maximum and minimum values if applicable, whether or not this attibute is inherited from the super class of this class, and finally its documentation. All the attributes together with the class name define a unique record.

In addition, the following two other tables provide additional information or serve as a link between tables.

- Class-Attribute table keeps cross-references between Class and Attribute tables.
- AttributePossibleValues table stores the possible values that an attribute can take. For example, "F" and "M" are the possible values for gender.

2.3 Interaction between the User Interface and the Database

Each visual component in the screens is associated with a component called DataExpress. Each of these DataExpress components contains the result of a query to the database. JBuilder's DataExpress provides support for direct data binding to dbSwing components. This kind of binding can be done with a simple setting of property in the component's Inspector. Whenever a DataExpress component changes, its corresponding visual component is automatically updated. After the system captures each input or modification from users, it updates the respective DataExpress component. Other DataExpress components and their associated visual components that are affected by the input are also updated. However, the data that are temporarily stored in the DataExpress components are actually saved into the database when the user presses the button. The JDataStore driver was used to access data in the JDataStore database.

In addition to the normal database visual component, the dbSwing library includes interactive navigation components. With this feature, users can enter values to locate data when the application is running. For example, the attribute list was implemented as a jdbNavList. The text box for the class that this attribute belongs to is associated with the same data query set. When the user clicks on an element in the attribute list, the data query set is also rolled to the corresponding record, and the text box for class is automatically updated. Hence, in this case, little programming is needed to update the text box.

The automated knowledge acquisition tool discussed in this paper can be used for documenting and storing knowledge items elicited on a problem domain. The tool is currently under development, and progress on the implementation process will be reported in the future.

Acknowledgements

The authors wish to acknowledge the generous support of a Strategic Grant from the Natural Sciences and Engineering Research Council of Canada. Contributions from current and former students of the Energy Informatics Laboratory of the University of Regina, Saskatchewan, Canada are also acknowledged.

References

[1] C.W. Chan, "Knowledge Acquisition by Conceptual Modeling", *Applied Mathematics Letters Journal*, vol. 3, p.7-12, 1992.

[2] C.W. Chan, "Development and Application of A Knowledge Modeling Technique", *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 7, no. 2, p.217-236, 1995.

[3] C.W. Chan and M. Johnston, "Knowledge Modelling for Constructing an Expert System to Support Reforestation Decisions", *Knowledge-Based Systems Journal*, vol. 9, p.41-59, 1996.

[4] C.W. Chan, "A Knowledge Modelling Technique and Industrial Applications", invited book chapter, Chapter 34 in Volume 4 of C. Leondes, Ed., *Knowledge-Based Systems Techniques and Applications*, 4 volumes, Academic Press, USA, 2000

[5] R.A. Flores-Mendez, P. van Leeuwen and D. Lukose. "Modeling expertise using KADS and MODEL-ECS", *Proceedings of Banff Knowledge Acquisition Workshop '98*, Banff Canada, October 1998.

[6] T. Gruber, "A translation approach to portable ontology specifications", *Proceedings of the seventh Banff Knowledge acquisition Knowledge-Based Systems Workshop '92*, Banff Canada, paper no.12, October 11-16, 1992.

[7] J. McDermott, "Preliminary steps toward a taxonomy of problem-solving methods", in: Marcus, S. (ed.), *Automating Knowledge Acquisition for Expert Systems*, Boston, MA: Kluwer Academic, pp.225-255, 1988.