Chapter XV
A Tool for Fuzzy Reasoning and Querying

Geraldo Xexéo
Universidade Federal do Rio de Janeiro, Brazil

André Braga
IBM Brazil, Brazil

ABSTRACT

We present CLOUDS, which stands for C++ Library Organizing Uncertainty in Database Systems, a tool that allows the creation of fuzzy reasoning systems over classic, nonfuzzy, relational databases. CLOUDS can be used in three flavors: CLOUDS API, a C++ API; CLOUDS-L, a compiled language; and CLOUDSQL, a fuzzy extension to SQL queries (ANSI, 1992). It was developed using the object-oriented paradigm and has an extensible architecture based on a main control system that manages different models, and runs queries and commands defined in them. As a test, it was incorporated into a geographic information system and used to analyze epidemiological data.

CLOUDS: TOOLS FOR FUZZY REASONING AND QUERYING

This chapter describes CLOUDS (C++ Library Organizing Uncertainty in Database Systems), a set of tools that allows a programmer to create or extend a database-based system with a fuzzy query engine provided fuzzy reasoning capabilities. It also describes its first real life application, the extension of an epidemiological geographic information system, GISEpi (Nobre, Braga, Pinheiro, & Lopes, 1997). CLOUDS is open source and can be downloaded from SourceForge.

Real world data are seldom as correct, exact, well defined, or well understood as our relational databases lead us to believe. Typically, we use approximation or intervals to deal with information uncertainty, often in a natural and unconscious way and at times with a clear loss of semantics. Fuzzy sets and fuzzy logic are well-established theories used to represent uncertainty in control systems (Klir & Yuan, 1995). Database researchers have also used them to model different forms of information uncertainty, creating fuzzy databases. Galindo, Urrutia, and Piattini (2006) provide a good review of different fuzzy database models.
A Tool for Fuzzy Reasoning and Querying

One of the most promising applications of fuzzy databases is representing uncertainty in geographic information systems (GIS) (Bosc, Kraft, & Petry, 2005), which are plagued by different imperfections in data, like imprecise representation of terrain or statistical errors on data acquisition (Bolstad, 2005). GIS are an important decision making tool for governmental, nongovernmental, and private organizations (Jankowski & Nyerges, 2001). To fulfill this role, they can be extended with a decision support system, possibly with a rule-based decision system based on knowledge collected from human experience. Again, fuzzy systems have established themselves as a good implementation strategy for rule-based decision systems, due to the capability of implementing human reasoning, including characteristics such as imprecision in data evaluation or simultaneously applying different rules (Pedrycz, & Gomide, 1998).

Motivated by the faulty data found in a real world health-care GIS application (Nobre et al., 1997), due to the quality problems in data gathering and compilation, and the need to use imprecise judgments to implement, or not, governmental health-care policies, we decided to extend it with a fuzzy module supporting data analysis and decision making. The result is CLOUDS, a portable library that can be easily used in different database applications. The library, developed in C++, contains tools for processing fuzzy SQL queries, for describing different fuzzy models over a relational database and for defining rules used in a fuzzy inference engine. In the next section, we will give a short review of the main topic discussed in this chapter. The third section will introduce CLOUDS. The fourth section will describe CLOUDS-L in detail. The fifth section will describe its use in GISEpi. The sixth section will present the conclusions.

**BASIC CONCEPTS**

We assume that the reader is aware of the main developments in fuzzy sets and fuzzy logic. However, we would like to review a few basic concepts that are the basis of our proposal.

**Fuzzy Systems and Fuzzy Reasoning**

“Fuzzy systems” is a general term encompassing all kinds of systems that use, in some part of their architecture, a mechanism based on fuzzy logic or fuzzy set theory. The traditional implementation strategy requires a main fuzzy engine that is isolated from the nonfuzzy (crisp) part of the system by crisp to fuzzy and fuzzy to crisp converters. Crisp to fuzzy conversion is known as fuzzification or fuzzy encoding. Fuzzy to crisp conversion is called defuzzification or fuzzy decoding (Klir & Yuan, 1995).

Fuzzy propositions are logical statements that assume a fuzzy value. They can be conditional, qualified, or both, as well as simple. A simple, unconditional, and unqualified proposition states that a variable element belongs to a fuzzy set, as in “the age of x is old.” For a particular element, the degree of truth of the proposition is interpreted as the degree of membership of this element to the fuzzy set. In this way, any fuzzy proposition can be interpreted as a possibility distribution function that is equal to the membership function of the fuzzy set. A simple proposition p has the forms (Klir & Yuan, 1995) “p : V is F” or “p : V(i) is F” if it is important to discuss the individual element referred to by the proposition, as in “The age of John is old.”

A qualified fuzzy proposition is a simple fuzzy proposition modified by a fuzzy truth qualifier or a fuzzy probability qualifier. Conditional propositions discuss the implication of one fuzzy proposition from another proposition, as in “If age is old, then strength is feeble.” Conditional propositions are equivalent to fuzzy implications. Among other options, it is common for a fuzzy system to use rules, or fuzzy implications, to formally represent knowledge. Although a single type of rule does not exist, the if-then representation is a standard. A basic if-then rule can be written in the form:

\[
\text{IF } a_1 \text{ AND } a_2 \ldots \text{ AND } a_n \text{ THEN } b, 
\]

where \( a_i \), \( 1 \leq i \leq n \), and \( b \) are simple fuzzy propositions. Like in the standard expert system approach,
24 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: www.igi-global.com/chapter/tool-fuzzy-reasoning-querying/20361?camid=4v1

This title is available in InfoSci-Books, InfoSci-Database Technologies, Business-Technology-Solution, Library Science, Information Studies, and Education, InfoSci-Library Information Science and Technology. Recommend this product to your librarian: www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

A Distributed System for The Management of Fine-grained Provenance
www.igi-global.com/article/a-distributed-system-for-the-management-of-fine-grained-provenance/142071?camid=4v1a

Large-Scale Sensor Network Analysis: Applications in Structural Health Monitoring
www.igi-global.com/chapter/large-scale-sensor-network-analysis/85461?camid=4v1a

Selecting a Database Partitioning Technique
www.igi-global.com/article/selecting-database-partitioning-technique/51123?camid=4v1a

Integrity Constraints in an Active Database Environment
Juan M. Ale and Mauricio Minuto Espil (2002). Database Integrity: Challenges and Solutions (pp. 113-143).
www.igi-global.com/chapter/integrity-constraints-active-database-environment/7880?camid=4v1a