Integrating Information Technology Practice into Logic Courses

YEFIM KATS
NJIT, University Heights, Newark, NJ 07002, USA, yefimkatz@yahoo.com

Abstract

The recent explosive growth of information technology triggered the rapid proliferation of logic-based commercial software. From this perspective, it seems imperative to introduce carefully selected industrial software packages into standard Logic and Critical Thinking courses, thus explicitly linking logical theory with existing application areas such as logic programming or data management. This paper is intended to demonstrate how logic-based industrial software can be used in a specialized as well as broad-based logic course.

The use of computer software for logic courses can be traced at least to the 1950th when "Patrick Suppes introduced his program Valid into… Stanford's elementary logic course". (BARWISE & ETCHEMENDY, 1996).

At this time, a number of software programs have been developed, assisting instructors and students in introductory and advanced courses; among them: Tarski's World for teaching first order logic, natural deduction based program Fitch, Turing's World, and Gateway to Logic - a European collection of elementary as well as advanced logic programs[1].

However, the software in question can be used only and only for the purpose of teaching logic - the links to the possible application areas are rather distant. In this paper we propose a completely different computer assisted approach to teaching Logic and Critical Thinking courses. This approach by no means is intended to substitute the already existing practice in this area. Our objective is to consider how logic-based industrial software can be explicitly introduced into standard logic courses taught by virtually every philosophy department around the world, thus providing students with the practical skills they could use pursuing a career outside the academia. The paper is intended as a brief introduction to a new teaching methodology rather than a detailed treatment of the vast area of logic software in education.

Among the relevant industrial information technology fields are Database Management, Logic Programming, and Data Mining.

As our first example, we consider the standard topic of quantification. (COPI & COHEN, 1998) The discussion of quantification theory in Logic courses can be integrated with - or even taught in a framework of - (tuple) relational calculus proposed originally by Codd (1972) as a logical query language for increasingly popular type of industrial software - relational database management system.

Tuple relational calculus is based on the standard first order calculus and intended to query industrial databases based on the relational data model; among them are such widely used products as Oracle, Microsoft SQL Server, and the dominant microcomputer database management system Microsoft Access - a part of Microsoft Office software package installed practically on any PC.

Relational calculus could be integrated into the standard Logic course in the following way. First, the students should be taught the standard predicate calculus and only then introduced to its applied modification - tuple relational calculus. Consequently, students may be asked to design (or to be provided with) a sample database, using one of the available commercial products. Once the database is in place, students can practice data manipulation queries using first, the natural language queries and then translating them into the relational calculus queries. At a next, more practical, phase students may be required to express calculus queries in the standard industrial query language SQL. The order as well as specific details of this process can vary and are up to a logic instructor.

Let's consider a simple example. Assume the following database schema:

Airline (carrierName, planeNo, destination)
Seat (seatNo, planeNo, seatType, price)

Here 'Airline' and 'Seat' are names of the tables created in any familiar to the student relational database system such as Microsoft Access. It's instructive to understand that in the logical context of relational data model a table is actually a name of a *relation*, while a column represents an *attribute* of a relation. The structure of a relation is often called by the familiar logical term *intension*; a set of all rows or tuples of any table represents an *extension* of a relation.

These somewhat technical logical terms of intension and extension could be linked to the standard introductory discussion of Socrates' contribution to philosophy. Socrates, as reported by his contemporaries, used to walk around Athens asking its citizens pointed questions like 'What is beauty?' or 'What is piety?' Usually all he could get in response was a definition by example: his opponents would simply give him a list of entities they would consider beautiful or a list of actions they would call pious. Now, such a list represents what we denote in logic by the term *extension*. However, it's not what Socrates wanted. He used his famous ironic criticism to show the listeners the inadequacy of the proposed definitions by extension; Socratic criticism was intended to demonstrate that such definitions, in modern logical terms, were either incomplete or inconsistent. What Socrates really insisted upon was an *essential definition* or a definition by *intension*, so that for each object presented we could tell whether a particular object possesses a certain property such as beauty or piety. (COPI & COHEN, pp. 137-148).

The point we want to make here is a methodological one. In a standard Introduction to Philosophy course, especially taught to technology or business majors, it may be instructive to show the connection between the moral inquiries of the old Athenian and the issues discussed in Logic or/and in the professionally oriented Database Management courses. By the same token, students could appreciate an unexpected leap from the 'dry' logic material to the poisoning moral deliberations in ancient Athens.

Let's return to our database schema. Following the outlined above methodology we can ask students to perform a sequence of queries on our database. Let's consider as an example the following query:

Show the air carriers with ticket price less than $1000

The students could be asked to translate such a query into the relational calculus query using logic notation. The corresponding calculus query would look like:

{ A. carrierName | Airline (A) ^ (E S (Seat (S) ^ (A.planeNo = S. planeNo) ^ (S.price <1000)) }

Where 'A' and 'S' are tuple relational variables, 'E' is an existential quantifier, and '^' stands for logical 'and'. As a reverse exercise, the students could be also presented with the calculus query and asked to restate it in an ordinary language.

The next important task – to present the above query in the commercial relational query language SQL:

```
SELECT carrier_Name
FROM Airline A, Seat S
Where A.planeNo=S.planeNo AND price < 1000
```

Why is this step so important for our methodology? - Because the SQL queries could be consequently run using Microsoft Access or another relational database management product. To do so, students would have, first, to implement the suggested database schema and, second, to create the schema *extension*. In practice, this procedure implies creation of the database as a file in Microsoft Access, creation of the tables within such a file, and finally requires populating the newly created tables. What is at stake here is an ability of an instructor to demonstrate the intrinsic connection between logical theory and information technology practice.

Following the outlined learning practice, philosophy majors would be able to 'feel' how the somewhat abstract logical constructions, which may be used by the philosophers in ontological discourse, play at the same time a significant role in the development and use of cutting edge information technology products; whereas information technology or business majors would be able to see how the seemingly hands-on information technology tools are based on deep logical results.

Another related area of interest for logic instructors is Data Mining - technique for finding hidden and unexpected patterns and relationships in sets of data. As an example, *manual data mining* may be used to illustrate the notions of *necessary and sufficient conditions* in logical reasoning or to demonstrate how logical operators can be used in query languages like relational calculus/SQL to discover implicit relationships in a data set.

[1] Available at http://logik.phl.univie.ac.at/~chris/formular-uk.html; a comprehensive list of logic software can be found at http://www.cs.otago.ac.nz/staffpriv/hans/logiccourseware.html

Let's consider the following 'disease diagnosis' schema adapted from R.J. Roger and M.W. Geatz (ROGER, & GEATZ, pp. 16-17, 2003):

Patient ID#	Sore Throat	Fever	Swollen Glands	Congestion	Headache	Diagnosis
1	Yes	Yes	Yes	Yes	Yes	Strep throat
2	No	No	No	Yes	Yes	Allergy
3	Yes	Yes	No	Yes	No	Cold
4	Yes	No	Yes	No	No	Strep throat
5	No	Yes	No	Yes	No	Cold
6	No	No	No	Yes	No	Allergy
7	No	No	Yes	No	No	Strep throat
8	Yes	No	No	Yes	Yes	Allergy
9	No	Yes	No	Yes	Yes	Cold
10	Yes	Yes	No	Yes	Yes	Cold

The students' task is to determine what sets of symptoms constitute the *necessary and sufficient conditions* - the standard topic for Logic courses - for a particular disease such as Strep Throat. (COPI & COHEN, p. 499). Following the standard method of scientific discovery an instructor may offer students a *hypothesis* (derived possibly from past experience):

The symptom of Swollen Glands is a necessary and sufficient condition for the diagnosis of Strep Throat

Then, students' task is to formulate a query, showing that aforementioned symptom constitutes a necessary condition for Strep Throat. The requested query could look like:

List all patients with diagnosis Strep Throat and no Swollen Glands

Following the methodology discussed in a context of database management, students should be able to translate the above natural language query into the relational calculus query and consequently into SQL query such as:

SELECT Patient_ID#
FROM Patients
WHERE Diagnosis='Strep_Throat' AND Diagnosis !='Swollen Glands'

When run in a commercial database system, the result of the query would be an empty table allowing us to conclude that Swollen Glands *is* a necessary condition for the diagnosis in question. The same procedure could be applied to find a sufficient condition - we leave this as an exercise to the reader. This somewhat simplistic example is intended to emphasize further the connection between logical aspects of data management on the one hand, and those of data mining on the other.

The important educational objective to keep in mind is to ensure that the progress in *theoretical* study of logic proceeds hand in hand with the progress in learning the *practical* applications of logic - data management, data mining or some other applied area such as logic programming.

The discussion of Data Mining could be extended to present database management and data mining as two aspects of a broader process known as knowledge discovery in databases (KDD). Consequently, KDD could be used as an industry oriented illustration of the general methodology of scientific enquiry. Such a comparison would incorporate a variety of topics such as the necessary and sufficient conditions, different forms of induction as well as probabilistic inference. (COPI & COHEN, CH. 13-14).

The topic of induction as well as different aspects of data management could be also introduced in the context of logically oriented programming languages such as Prolog and Datalog (ULLMAN, WIDOM, 2002; BRATKO, 2001). The acquired, in this way, elementary skills in logic programming would be helpful to expose students to the vast area of Artificial Intelligence (AI) - still another field where logic, philosophy, and industry merge. In particular, AI based Decision Support tools would allow an instructor to add additional value to the discussion of professional reasoning in almost any area – from medicine to finance. In this context, several interesting software packages are available from Banxia Co., including Decision Explorer described as "a tool that has been designed to help you to see relationships between different ideas and perspectives which might be expressed about any subject"[2].

The exact scope as well as the level of practice-oriented Logic and Critical Thinking courses should be determined by each college, depending on its orientation as well as faculty and students' interests. However, the general methodology focused on the proposed integration of modern industrial software into logic curriculum is definitely worth considering.

BIBLIOGRAPHY

BARWISE, J. & ETCHEMENDY, J., "Computers, visualization, and the nature of reasoning", 1996, accessed at http://www-csli.stanford.edu/hp/index.html.

BRATKO, I. *Prolog: Programming for Artificial Intelligence*, 3rd ed., Addison-Wesley Pub., 2001.

COPI, I.M. & COHEN, C., *Introduction to Logic*, PH, 1998, 10th ed. ch. 10.

ROGER, R.J. & GEATZ, M.W., *Data Mining: A Tutorial-Based Primer*, Addison Wesley, 2003.

ULLMAN, J.D., WIDOM, J., *A First Course in Database Systems*, Prentice Hall, 2002.

[2] See www.banxia.com

Education for adults in Russia: steering and promoting at the regional level

ELLA ISKRENKO

Volgograd State University 100, Prospect Universitetsky
400062 Russia, fno@volsu.ru

Abstract

The paper analyses the practice that Volgograd State University has had in the development of business education for adults through international projects. The author describes international partnership experience that helps build up effective dialog and relationship among the regional business community, higher education and employment authorities, employers and other higher education stakeholders who strive for cooperation in the design and delivery of continuing education programs.

CONTEXT

At present Russia is experiencing a substantial gap between the needs of the national economy and training of certified specialists at conventional higher education institutions. Therefore the two main tendencies of the higher school development in Russia over the recent years have been the growing number of retraining programs on the one hand, and universities' efforts to enhance the quality of educational services that such programs provide on the other hand. The objective of the traditional training of "diploma specialists" in our country comes in the foreground, but there is also the task of raising the skill level of the national workforce through an efficient retraining system for those who already hold higher education degrees.

Russia's transition to market economy created a huge need for quality business education. Universities began to lay emphasis on competitiveness, high quality and choice of the most effective educational projects. In order to solve the task of creating the system of business education in Russia, they thoroughly studied the progress of the best Western business schools. It is not uncommon when Russian universities' professors and lecturers are trained at Harvard, Massachusetts Technology Institute, London Business School, and other prestigious higher education institutions of the world. Creative application of the world experience allows them to initiate short and long-term joint educational projects with the leading Western business schools. There is a big variety of such projects and Russian higher education institutions are becoming more and more inventive in designing them to meet the demands of their services' consumers.

The international partnership projects in the sphere of business education provide considerable help in the organization of business skills training.