University of Nebraska - Lincoln
DigitalCommons@University of Nebraska - Lincoln

CSE Conference and Workshop Papers

Computer Science and Engineering, Department of

12-9-2003

Spatiotemporal databases: Models for attracting students to research

Ágnes Bércesné Novák
Pazmany Peter Catholic University, bercesne.novak.agnes@itk.ppke.hu

Peter Revesz
University of Nebraska-Lincoln, prevesz1@unl.edu

Zsolt Tuza
University of Pannonia, tuza@dcs.uni-pannon.hu

Spatiotemporal Database Models for Attracting Students to Research

Ágnes Bércesné Novák

Budapest Polytechnic, John von Neumann Faculty of Information Technology

bercesne.novak.agnes@nik.bmf.hu

Peter Z. Revesz

University of Nebraska Lincoln, Department of Computer Science and Engineering, USA

revesz@cse.unl.edu

Zsolt Tuza

Computer and Automation Institute Hungarian Academy of Sciences and

University of Veszprém

tuza@sztaki.hu

Abstract*: In higher education professors often make much effort to introduce their students to research. Unfortunately, the present standard database systems curriculum is composed of well-settled subjects that do not lead to research. The challenge is to bring the research frontier closer to students at beginner level. In this paper we describe how it can be done in the area of spatiotemporal databases. We propose a new database systems curriculum and illustrate its benefits by mentioning several highly succsesful student projects in some recent experimental introductory database systems courses that followed the new curriculum.*

Keywords*: constraint databases, spatiotemporal data, relational database, logical representation, teaching*[1]

---

[1] A.B. Novak, P. Z. Revesz, Z. Tuza, Spatiotemporal databases: Models for Attracting students to research, *Proc. Int. Conference in Memoriam John von Neumann,* John von Neumann Computer Society Press, pp. 247-256, Budapest, Hungary, Dec. 2003.

1 Motivation and introduction

John von Neumann, who is famous in computer science for his revolutionary invention of the von Neumann Computer Architecture, once said that: "If people do not believe that mathematics is simple, it is only because they do not realize how complicated life is." [1] Although few would agree with von Neumann's opinion entirely, it is true that good teaching can greatly simplify any material and make students interested early on in research. In particular, the area of spatial and spatiotemporal databases offers many new and exciting problems, which can be modeled and implemented by 2nd and 3rd year students. The main advantages of this area are that (1) the topic has many applications in transportation, telecommunication, weather forecasting etc., (2) the problems are always nicely visualizable by computer animations, maps, and graphs.

We organize the rest of this paper as follows. In Section 2, we first briefly review the necessary parts of the relational, spatial and temporal databases. We focus only on that part of the theory, which can be adapted for a first course. However, we want to emphasize that in each area there are complex applications and also deep theoretical results. What makes this interesting is exactly the fact that almost any interested person can contribute with some real application or modeling, or theoretical investigation of simpler problems.

In Section 3, we outline the current Hungarian first course standard on database systems. Then we compare this standard with the one taught in at the University of Nebraska-Lincoln, in the United States. We also describe the opportunity of inserting the new concepts into a base level course.

In Section 4, we list the students' results of a course held in the summer of 2003 by the first author while visiting the University of Nebraska-Lincoln, and students' theses based on supervised independent learning at the Budapest Polytechnic University.

Finally, in the last section, we close this paper with a few concluding remarks.

2 Spatial, spatiotemporal relational models and constraint databases

The database area continues to grow so rapidly, that it is impossible to review all the theoretical and technical aspects. Hence, we restrict ourselves only the very basic notions for the relational, spatial and temporal relational models. The latter two were introduced in higher education only in the past few years. Sections 2.2.2 and 2.2.3 are based on [8].

2.1 Relational models

The relational model was invented first by Codd, for finite relations [2].

The model is based on the mathematical relation r: it is a subset of a direct product of some given domains D_n:

$r \subseteq D_1 \times D_2 \ldots \times D_n := \{(d_1, d_2, \ldots d_n) \mid d_k \in D_k \; 1 \leq k \leq n \}$

Using this idea, relations and also relationships between relations can be modelled in a unified way. A relational database is a set of relations, i.e. the instances of them. The description at the logical level is simply the collection of the relation schemes and the prescribed constraints. A relation scheme is the description of the context of the relation, ie. the notions corresponding to the given domains. As relation instances can be imagine as tables, so the schema of a relation can be visualize as the header of that table. For example LINE(POINT1, POINT2) is a schema of a relation containing supposedly lines going through points. However, the actual data structure is not defined here: we do not know, whether these points are represented by their coordinates or just by an idenfier. That is why it is called the logical level of abstraction. At that level one does not care about the physical storage problems, even not with variable types. The domains (ie. here POINTk) in the logical schema are called attributes. In the schema the list in the brackets is the list of attributes. One row (or tuple) in a relational instance (a table) belongs to the relation if it is explicitely stored in it. It also means, that in this model only a finite number of tuples can be stored. However this simple, obvious assumption will be changed in the case of constraint databases.

2.2 Spatial models

There are several spatial (geometic) models, e.g. the so called Raster-, Spaghetti-, Peano- and Polynomial Model, which are not discussed here, because of the lack of space. Let us note, that for example the 2-Paremetric Spaghetti Model (also the temporal extension) can be trasformed easily to the constraint model, described in 2.2.2. The interested reader can found a brief introduction into these models in [3] and also other spatial models and practical applications in [4]. Spatiotemporal databases and related topics are thoroughly discussed in [8].

2.2.1 Storing topological information: the PLA-model

The US Bureau of the Census described the PLA-model first in [5]. This model is very suitable to represent topological information. That is, the actual physical situation is not represented in this model. Originally it is only for planar graphs (maps). The authors of [6] began a systematic investigation of this model, and extended the original one for directed and weighted cases in [7]. The original schema of the database is:

R1(LINE-ID, FROM, TO): In this relation the edges of the graph are stored by their vertices.

R2 (LINE-ID, LEFT-AREA, RIGHT-AREA): As the names indicate, in this relation the areas of the different sides of the temporarily oriented edges can be found. Orientation is needed in order to get the infinite area.

R3(AREA-ID, POINT-ID, LINE-ID, LABEL): In R3 the borders of the areas are stored, which is an alternate sequence of points and lines

R4(POINT-ID, LINE-ID, AREA-ID): The role of this relation is to represent the environment of a vertex: it is an alternate sequence of lines and areas.

It is proved in [6] that R3 alone is complete, all topological information about the stored graph can be derived from it, involving the other three relations.

2.2.2 Storing geometrical information: constraint (relational) databases

As we mentioned in 2.2.1, the relational model was developed to store only finite relations. But in case of spatial data we usually need to store infinitely many points. To overbridge this difficulty, a new abstraction level, the constraint level was introduced. This level of abstraction allows us to represent infinitely many tuples in a finite way. Constraint databases were introduced in [9].

A constraint database is a finite collection of constraint relations. A constraint relation is a finite set of constraint tuples. The infinity is "stored" through the constraints. It is built up from atoms, for example the arithmetic atomic constraints are as follows:

Equality:	$u = v$,	Inequality:	$u \neq v$
Lower bound:	$u \geq b$,	Upper bound:	$-u \geq b$
Order:	$u \geq v$,	Gap-order:	$u - v \geq b$ where $b \geq 0$
Difference:	$u - v \geq b$,	Half-Addition:	$\pm u \pm v \geq b$ where $b \geq 0$
Addition:	$\pm u \pm v \geq b$,	Linear:	$c_1x_1 + \ldots\ldots + c_nx_n \geq b$

Polynomial: $p(x_1,\ldots\ldots,x_n) \geq b$

Of course, instead of $\geq$ we can also use $>$.

Constraint formulas can be generated in a usual way ; each atomic constraint is a formula, and if F and G are formulas, then (F and G) is a formula and (F or G) is a formula. Also if F is a formula, then (not F) is a formula as well. Now the tuples can be interpreted as if $F(x_1,\ldots,x_n)$ is formula with n variables, $t = (c_1,\ldots,c_n)$ a tuple with n constants. Tuple t satisfies F, if substituting each x_i by c_i makes F true. That is, at the logical level we have infinitely many tuples in the relation, however we have finite number of constraint tuples in a constraint relation at the constraint abstraction level. Of course, the idea can be applied not only for the geometic data. As an example, see the schema: Taxtable(Income,Tax). In case there is no upper bound for attribute Income, the Taxtable relation would be infinite. Introducing constraints, one of the constraints could be:(i, t : $0 \leq i$, $i \leq 24000$, $t = 0.15$ i). In other words, (i, t : $0 \leq i$, $i \leq 24000$, $t = 0.15$ i)is a constraint tuple of Taxtable(Income, Tax). It means that the tax is 15% on any income below $24,000. It can be pictured as if the header of the relation would be Taxtable(Income, Tax, Constraints). It "contains" the pairs (value1, value2) through substitution, satisfying the constraint.

2.2.3 Storing temporal information: the spatiotemporal constraint (relational) model

Here we also do not go into deep details: instead of reviewing the exisiting kinds of data models, we restrict this description to the main idea only. But let us at least comfort the reader, that most of the models, like Extreme Point Data Models (e.g. rectangle data model, Worboys' data model), Parametric Extreme Point Data Models (parametric rectangles and parametric 2-spaghetti data models) and also Geometric Transformation Data Models, etc. can be translated into constraint databases, as you can find in Chapter 13 of [8].

Spatiotemporal objects have both spatial and temporal extents. The spatial extend of an object are the points in space belonging to the objects. The temporal extent is the time inerval, when these points were/are exisiting. Both extent can be formulated by the constraints listed in 2.2.2. A spatiotemporal schema is e.g. Window(id, x, y, t), where the first attribute correspond to an open window identifier on a computer screen, *x, y* are spatial points of the window, and *t* is the time when it is active. As from this example the reader may conclude, in the case of this spatiotemporal model, the underlying data model is really the constraint spatial model, just one of the dimensions is interpreted as the time. But this "just" means a lot: for example through this variable the animation goes wonderfully smoothly, which is very attractive for the students.

3 Spatial and spatiotemporal databases in higher education

In this section we briefly summarize that part of the curricula dealing with Information Systems, into which we think to insert a minimum material on spatial and spatiotemporal relational models. This mimimum allow the students to join to the recent research areas. For simplifying terminology, from now on, we briefly refer to the models reviewed in the previous chapters as extended relational models.

3.1 Experiences of teaching extended relational models

At the University of Nebraska-Lincoln (UNL) a new course was developed in the academic year 2002/2003. The previous standard can be called a classsical one. Over the years, in the international database community the introductory courses became very similar to each other, commonly containing material such as the entity-relationship model, the relational model, the relational algebra, functional dependencies, normalization based on functional dependencies, 1-3 Normal forms, Boyce-Codd Normal Form, and SQL queries. This is the minimum we can see in the course descriptions all over the world. Some courses (also at UNL) contain additional basic knowledge on transaction management and recovery systems.

Until 2002/2003 the UNL database course used [11] as the textbook. However, in the fall of 2002 the standard curriculum on Database Systems was greatly extended by the coverage of various spatiotemporal data models and systems, and in the spring of 2003 the course was devoted almost entirely to these new aspects, primarily based on [8].

In the summer of 2003, the database systems course (held by the 1st author) was also the classsical standard but enhanced by the constraint model and a project on it. There were three lectures per week and the students carried out the laboratory assignments independently.

In all of these courses the MLPQ/PReSTO system [10] was entirely or partly used. This system was developed and implemented at UNL, at the Department of Computer Science and Engineering, and it can be downloaded free for education purposes from the second author's web-page [10]. This system can be used in various ways. It has the usual database engine for standard SQL, a Datalog module and it also is capable of running SQL and Datalog queries on both relational and constraint/spatiotemporal databases. The system has also an automated built-in animation facility. for constraint/spatiotemporal databases. The system has an automated built-in animation facility, too.

3.2 Opportunities to teach extended relational models in the Hungarian higher education

Browsing among the curriculas in the public web-pages of the leading Universities in Hungary, such as Budapest University of Technology and Economics, Loránd Eötvös University (Budapest), Kossuth Lajos University (Debrecen), Attila József University (Szeged), University of Veszprém (Veszprém), Budapest Polytechnic (Budapest) etc. one can recognize that the Hungarian academic community also follow the “classical standard” regarding database courses. Without exception, the relational model and at least one of its query languages are taught. Normalization, based on the theory of functional dependencies is also in the base course. In some universities we can find material on basic transaction handling, recovery and Datalog as well. The software used is usually only one of the products offered by SQL implementer vendors.

At several universities, the instructors use their own lecture notes, however, mainly [12] (in Hungarian translation) is suggested as a complementary or regular coursebook. The number of classes per week varies from 2 lectures and 2 labs to a maximum of 4 lectures and 2 labs.

Comparing with the UNL curriculum, the main difference is hidden not in the theory, but rather in the applications. Even in the worst case, supposing 2 lectures a week and later labs, the main idea can be inserted and taught in a few lectures.

The ideal place for it is the relational model, relational algebra. The generalization of the relational model into the constraint model for the students come naturally. Examples, without changing the usual material too much, helps to teach in an efficient way. E.g. relational algebra also can be introduced by minor efforts, as the example from the UNL course in Figure 1 shows.

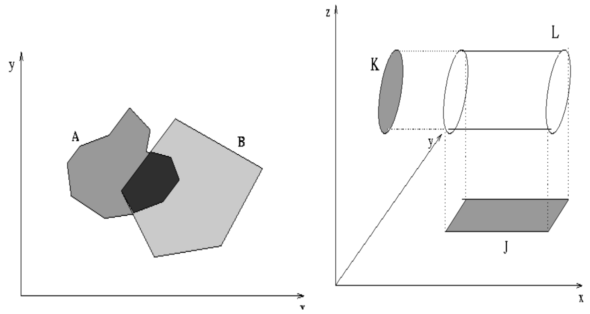

Figure 1: Teaching relational algebra through spatial models: area A intersection area B and natural join of area K and area J are illustrated.Adopted from [8]

The main modification, which is needed, would be in the laboratory work, developing the projects in the MLPQ/PReSTO System. However, the system can also be used in a standard way, for evaluating SQL queries only, it would be useful to teach DATALOG basics. The MLPQ/PReSTO system could also be useful in the discrete mathematics courses, where SQL can be mentioned as an example of applying logic as a query language used in that and other database systems.

The PLA model, involving also the weighted and directed extended cases, it can be taught as an example of a possible graph representation, different from the usual storing methods. It is also connected to discrete mathematics.

As outlined above, we believe that with minor changes in curricula and a little effort in teaching we can achieve a basic, but applicable level of knowledge on the extended relational models even at the beginners level.

4 Students' work

In this section we mention some of our students' work. In the first part, we briefly review the results of the Hungarian students, next we give some examples of the projects of the U.S. students.

4.1 Students' results in studying the PLA model

In the Budapest Polytechnic the extended models are not taught in the regular courses. However, each student, choosing PLA databases for his/her thesis was trained individually by the supervisor. They were very enthusiastic and developed themselves not only in the strict database topic, but also in the corresponding discrete mathematics area.

In the area of software development, a DELPHI implementation on producing a linear time planar embedding of a graph stored in R3 can be mentioned [12].

New algorithms for finding

- Eulerian and bipartite graphs
- Spanning tree, minimal spanning tree
- Adjacency matrix, incidence matrix from PLA

are described in [13], [14], [15] respectively.

4.2 Students' results in studying spatiotemporal databases

The first examples are restricted only to the Summer Course in 2003, where the students learnt the corresponding theoretical material only in 3 times 2 lectures, and had an introductory lab session for 4 hours. The nicest projects involving animation are as follows:

- smiling faces
- bird feeding
- map changes in time
- bus time-schedule

In the Spring and Fall 2003 courses at the University of Nebraska-Lincoln, the students were exposed more to the spatiotemporal material as we described above, and the student projects were also more advanced in general. The students not only used animation, but used other advanced visualization tools, such as map displays with meaningful color bands, and some 3D displays. In addition, students also wrote quite advanced SQL and Datalog queries and made their databases web accessible. The web pages provided users a click-able menu of all their already implemented SQL and Datalog queries.

Towards the end of the course many research topics were mentioned. In fact, from the spring 2003 course, a group of five graduate students continued in the summer to do some research work in the database area in independent study courses supervised by the second author. Two M.S. theses and one Ph.D. thesis is expected to be completed in the area by the end of December 2003 [17], [18], [19].

Conclusions

The aim of this paper is to show that students' research work can be encouraged in the topic of the extended relational models. One can check on the internet and other resources that this topic is really hot, hence our students will not be bored by dealing with ancient problems.

Although the present Hungarian teaching curriculum adheres closely to the "classical standards" and does not cover even the basic knowledge in the spatiotemporal database area, the students can be given an exciting experience in research even before completely finishing their studies in the database systems area.

Our proposed curriculum changes were already tested and proved successful in the USA as illustrated by the above mentioned student projects.

Also the PLA examples show that even at the application level worthy results can be obtained, in the present circumstandes.

In summary, the authors hope, that this paper illustrates one other idea of von Neumann, namely that the efficiency of a system (in our case a small learning and research group at a university) mainly depends on the intercommunication between the units, from which the system is made up (see [20], [21]).

References

[1] Edit Kádár: John von Neumann was born 100 years ago, Magyar Nemzet, 2003. 04.11., (in Hungarian)

[2] E. F. Codd: A relational model for large shared data banks, Communication of the ACM, 13 (6), pp 377-87, 1970.

[3] Jan Paradaens: Spatial Databases, The Final Frontier, Lecture Notes in Computer Science, 893, Database Theory-ICDT'95, pp 12-32, 1995.

[4] R.Laurini, D. Thompson: Fundamentals of Spatial Information Systems, APIC Series, 37 , Academic Press, 1992.

[5] J. P. Corbett: Topological Principles in Cartography, Technical paper 48,US Bureau of the Census, Washington DC, 1979.

[6] Á. B. Novák, Zs. Tuza: Reconstruction Graphs and Testing Their Properties in a Relational Spatial Database, Computers and Mathematics with Applications 43, pp 1391-1406, 2002.

[7] Á. B. Novák, Zs. Tuza: Representing Directed and Weighted Graphs and Their Properties in a Relational Spatial Model, 6th Conference on Intelligent Engineering Systems, INES, 2002.

[8] P. Z. Revesz: Introduction to Constraint databases, Springer, 2002.

[9] P. C. Kanellakis, G. M. Kuper, P. Z. Revesz: Constraint Query Languages. Journal of Computer and System Sciences, 51, pp 26-52, 1995.

[10] P. Z. Revesz: MLPQ/PReSTO Users' Manual, http://www.cse.unl.edu/~revesz/MLPQ/mlpq.htm

[11] J. Ullman, J. Widom: A First Course in Database Systems, Prentice Hall, 1997., Hungarian language edition published by Panem Kft. 1998.

[12] G. Gaulandt: Graph Reconstruction from the PLA model, Thesis, Budpaest Polytechnic, Hungary, supervised by Á. B. Novák, 2001.

[13] Á. B. Novák, T. Fehér: Constraints for Bipartite and Eulerian Graphs in a Spatial Relational Model, Proceedings of the 4th International Conference on Intelligent Emgineering Systems, INES 2000.

[14] Á. B. Novák, Zs. Ronkay-Zs. Tuza: Discovering Spanning trees in a Spatial Relational Model, Proceedings of the IEEE International Conference on Computational Cybernetics, ICCC 2003 .

[15] Zs. Sajner: Representing some graph classes in the PLA model, Thesis, Budpaest Polytechnic, Hungary, supervised by Á. B. Novák, 2001.

[17] Y. Chen: Aggregation in Spatiotemporal Databases, Ph.D. thesis, University of Nebraska-Lincoln, 2003, supervised by P. Revesz, 2003.

[18] S. Anderson: Constraint Datalog in Trust Management, M.S. thesis, University of Nebraska-Lincoln, supervised by P. Revesz, 2003.

[19] V. Ramanathan: Experiments in Constraint Automata, M.S. thesis, University of Nebraska-Lincoln, supervised by P. Revesz, 2003.

[20] William Aspray: John von Neumann and the Origins of Modern Computing, MIT Press, Cambridge, Massachusetts, 1990.

[21] Norman Macrae: John von Neumann, Pantheon Books, New York, 1992.