# Topics for
# PhD Theses, Master Theses, and Individual Studies

**Markus Schneider**

Department of Computer & Information Science & Engineering (CISE)
Database Systems Research & Development Center
E450 CSE Building
PO Box 116120
Gainesville, FL 32611-6120
Tel: (352) 392-2697
Fax:(352) 392-1220
Email: mschneid@cise.ufl.edu

*September 12, 2002*

## Introduction

The topics for students' projects described below correspond to my current research interests which are in databases in general, in spatial databases, in spatio-temporal databases, in fuzzy databases, and in bioinformatics. For further information, please look at my web page *http://www.cise.ufl.edu/~mschneid/*. A specific and interesting topic a student has in mind is also welcome but has, of course, to be discussed with me and to be accepted by me.

In order to save the student's and my time, I expect from the student that (s)he has made himself (herself) a little bit familiar with the main aspects of the desired or required knowledge which is listed in each topic description. This means the student should have made a limited personal study with respect to the topic of interest and should (at least) have a rough idea what the topic is about. Deep knowledge in the listed areas is not required. But the student must demonstrate that (s)he is willing to acquire the needed knowledge. Later the student must show that (s)he is capable of coping with the topic of interest and that (s)he has understood the required background.

Please contact me then by email (*mschneid@cise.ufl.edu*) in order to make an appointment. Also, a student with an own topic in mind should write me an email. The email should contain the student's resume and give a motivation why (s)he is interested in one or several topics. If more explanations are needed concerning a topic (after the student's personal study!), please write me an email.

## General Procedure during a Project

Having agreed upon a topic, in a first phase, the student starts with a detailed study of related work. The result of this study is a written document which describes, *analyzes*, and *structures* the references found, identifies their benefits and drawbacks, and precisely lists the sources of the references at the end. The resulting document is the student's first deliverable and will be part of the thesis.

Having learnt from the literature study, in a second phase, the student starts with a conceptual design for solving the problems of the topic. This conceptual design proceeds hand in hand with me and is based on common discussions which take place approximately every three or four weeks in my office. The conceptual proposals have to be written down in a more or less detailed way by the student. They serve as the foundation of discussion between the student and me. If there is an agreement upon a concept, the student writes a preliminary "final" description about it which is a second deliverable for me and will later be part of the thesis. The reason for this is that at the time of designing a concept, the student is fully aware of its details and thus able to write them all down. If the student describes a previously devised concept to a much later point in time, (s)he will have forgotten many of the relevant details and motivations for the conceptual decisions.

Based on the agreed concept, in a third phase, the student devises an implementation method for it, that is, data structures and algorithms. This method is described in a document first, which is the third deliverable, before it is implemented in a programming language.

The next phases are iterations of the second and third phase. Concepts, implementation methods, and implementations are added and described in written form. It is highly recommended to include all written parts in the *same* document with a meaningful structure (explicit table of contents). This will later help you to find the gaps in your thesis, concept, and implementation.

In the last phase, the thesis is written or completed. The single document can serve as the basis for it. You in principle have to revise the document, add parts, and write transitions between different parts. The implementation has to be completed too. During the implementation phases, the code must be documented.

## Topics Overview

(A click on a topic number will lead you directly to a more detailed topic description.)

| | | | |
|---|---|---|---|
| *Topic 1* | Design and Implementation of a Type System for Two-Dimensional Crisp and Vague Spatial Data and its Integration into Database Systems | Master | available |
| *Topic 2* | Design and Implementation of a Type System for Three-Dimensional Spatial Data and its Integration into Database Systems | PhD,<br>Master | available |
| *Topic 3* | Implementation of a Type System for Spatio-Temporal Data and its Integration into Database Systems | Master | available |
| *Topic 4* | Extension of a Past-Oriented Spatio-Temporal Data Model to Near Future Evolutions | PhD,<br>Master | available |
| *Topic 5* | Implementation of Spatio-Temporal Predicates and Their Visual Specification | Master | available |
| *Topic 6* | Conceptual Modeling of Database Applications Including Spatial, Temporal, and Spatio-Temporal Aspects | Master | available |

| | | | |
|---|---|---|---|
| *Topic 7* | The Integration of Spatial Applications on PDAs | Master | assigned |
| *Topic 8* | Design and Implementation of a Type System for Two-Dimensional Crisp and Vague Spatial Data by Using the Functional Programming Language Haskell | Master | available |
| *Topic 9* | Metric Refinements of Topological Relationships | Master, Individual Study | available |
| *Topic 10* | A C++-Library for the Postscript Programming Language | Individual Study | available |
| *Topic 11* | Exploiting Data Compression Techniques for Processing and Querying in Database Systems | Individual Study | assigned |
| *Topic 12* | Redesign and Reimplementation of the CISE Department's Web Site | Master | available |

## Topics in Detail

*Topic 1:* Design and Implementation of a Type System for Two-Dimensional Crisp and Vague Spatial Data and its Integration into Database Systems

*Abstract.* This topic deals with the design and implementation of *two-dimensional spatial data types* like *point*, *line*, and *region* in the Euclidean plane, and their integration into extensible, full-fledged database systems. Although some spatial data type extension packages have already been implemented in the past, these still suffer from weaknesses like lacking generality, missing functionality, and non-existing numerical robustness. The spatial type system to be constructed is to include crisp and vague spatial data. *Crisp* spatial objects have a precisely determined boundary like countries and districts with their political and administrative boundaries or land parcels with their cadastral boundaries. *Vague* spatial objects do either not have sharp boundaries, or their boundaries cannot be precisely determined. Examples are land features such as population density, soil quality or vegetation, oceans, biotopes, deserts, an English speaking area, clouds, and sandbanks. For testing and demonstrating purposes, a graphical user interface for the visualization of 2D spatial objects and the evaluation of operations should be included. Another important aspect is to design mechanisms to integrate this type system into an extensible database system.

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* databases in general, spatial databases, 2D computational geometry; programming skills in C++

*Introducing Literature:*

Ralf Hartmut Güting. An Introduction to Spatial Database Systems. *VLDB Journal* (*Special Issue on Spatial Database Systems*), 7(3), 231-246, 1994.

Markus Schneider. *Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems*. LNCS 1288, Springer-Verlag, 1997.

Martin Erwig & Markus Schneider. Vague Regions. *5th Int. Symp. on Advances in Spatial Databases* (*SSD*), 298-320, 1997.

F.P. Preparata & M.I. Shamos. *Computational Geometry.* Springer Verlag, 1985.

*Topic 2:* Design and Implementation of a Type System for Three-Dimensional Spatial Data and its Integration into Database Systems

*Abstract.* So far, spatial databases have mainly dealt with the management of two-dimensional geometric data. This has turned out to be sufficient for many spatial applications. But on the other hand, our world is three-dimensional. In this project, first possible applications are investigated that require or would greatly benefit from a treatment of three-dimensional data and operations. This study should lead to an identification of the essential *three-dimensional spatial data types* and operations. In the following, the data types and operations are formally defined and implemented according to a number of design criteria posed before. Examples of such design criteria are generality of the data type definition, closure properties of data types, and numerical robustness. For testing and demonstrating purposes, a graphical user interface for the visualization of 3D spatial objects and the evaluation of operations should be included. Another important aspect is to design mechanisms to integrate these types into an extensible database system.

*Appropriate for:* PhD, Master

*Status:* available

*Desired or required knowledge:* databases in general, spatial databases, 3D computational geometry; programming skills in C++

*Introducing Literature:*

Ralf Hartmut Güting. An Introduction to Spatial Database Systems. *VLDB Journal* (*Special Issue on Spatial Database Systems*), 7(3), 231-246, 1994.

Markus Schneider. *Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems*. LNCS 1288, Springer-Verlag, 1997.

Aristides A. G. Requicha. *Geometric Modelling: A First Course.* 1995-2000. URL: http://www-pal.usc.edu/~requicha/book.html

Sara Anne McMains. *Geometric Algorithms and Data Representation for Solid Freeform Fabrication*. Dissertation, 2000. URL: http://www.me.berkeley.edu/~mcmains/thesis.pdf

*Topic 3:* Implementation of a Type System for Spatio-Temporal Data and its Integration into Database Systems

*Abstract.* Spatio-temporal database systems deal with geometries changing over time. If only the position in space of an object is relevant, then *moving point* is a basic abstraction. Examples are people walking around, animals migrating over time, cars driving in a city, or planes connecting cities or countries. If also the extent is of interest, then the *moving region* abstraction captures moving as well as growing or shrinking regions. Examples are storms traversing a country and at the same time altering their shape, forest shrinking, and high/low pressure areas or temperature zones changing dynamically. This leads to *spatio-temporal data types*, and spatio-temporal databases are essentially *moving objects databases*. A conceptual design for spatio-temporal data types has already been proposed. The remaining, important, and challenging task is to find an efficient and numerically robust implementation for it. For testing and demonstrating purposes, a graphical user interface for the visualization of spatio-temporal objects and the evaluation of operations should be included. Another important aspect is to design mechanisms to integrate these types into an extensible database system.

*Remark:* This topic forms the basis for Topic 4 and Topic 5.

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* databases in general, spatio-temporal databases, spatial databases, computational geometry; excellent programming skills in C++

*Introducing Literature:*

Martin Erwig, Markus Schneider, Ralf Hartmut Güting & Michalis Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica*, 3(3), 265-291, 1999.

Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos Markus Schneider & Michalis Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* (*TODS*), 25(1), 1-42, 2000.

Luca Forlizzi, Ralf Hartmut Güting, Enrico Nardelli & Markus Schneider. A Data Model and Data Structures for Moving Objects Databases. *ACM SIGMOD Int. Conf. on Management of Data* (*SIGMOD*), 319-330, 2000.

### *Topic 4:* Extension of a Past-Oriented Spatio-Temporal Data Model to Near Future Evolutions

*Abstract.* In spatio-temporal database systems it is not only interesting to manage and analyze past evolutions or movements of spatial objects, but also to be able to calculate in advance or even to forecast their *near future* motion and behavior. Examples are the route of a whale herd in the next hours or in the next two days, or the destructive way of a hurricane. The task is here to devise and implement concepts for near future developments of spatial objects. It is important that the database has not to be updated all the time, because this is very inefficient. The solution approach should result in a suitable extension of our data model for past-oriented spatio-temporal data and identify important, future-oriented operations and predicates. For testing and demonstrating purposes, a graphical user interface for the visualization of spatio-temporal objects and the evaluation of operations should be included. Another important aspect is to design mechanisms to integrate these types into an extensible database system.

*Remark:* This topic is based and dependent on Topic 3.

*Appropriate for:* PhD, Master

*Status:* available

*Desired or required knowledge:* databases in general, spatio-temporal databases, spatial databases, computational geometry; programming skills in C++

*Introducing Literature:*

A.P. Sistla, O. Wolfson, S. Chamberlain & S. Dao. Modeling and Querying Moving Objects. *Int. Conf. on Data Engineering* (*ICDE*), 422-432, 1997.

O. Wolfson, B. Xu, S. Chamberlain & L. Jiang. Moving Objects Databases: Issues and Solutions. *10th Int. Conf. on Scientific and Statistical Database Management* (*SSDM*), 111-122, 1998.

O. Wolfson, S. Chamberlain, S. Dao, L. Jiang & G. Mendez. Cost and Imprecision in Modeling the Position of Moving Objects. *Int. Conf. on Data Engineering* (*ICDE*), 588-596, 1998.

O. Wolfson, A.P. Sistla, S. Chamberlain & Y. Yesha. Updating and Querying Databases that Track Mobile Units. *Distributed and Parallel Databases*, *7*, 257-387, 1999.

A.P. Sistla, O. Wolfson, S. Chamberlain & S. Dao. Querying the Uncertain Position of Moving Objects. O. Etzion, S. Jajodia, and S. Sripada (eds.), *Temporal Databases - Research and Practice*, LNCS 1399, 310-337, 1998.

O. Wolfson, L. Jiang, A.P. Sistla, S. Chamberlain, N. Rishe & M. Deng: Databases for Tracking Mobile Units in Real Time. *7th Int. Conf. on Database Theory* (*ICDT*), 169-186, 1999.

Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos Markus Schneider & Michalis Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* (*TODS*), 25(1), 1-42, 2000.

## *Topic 5:* Implementation of Spatio-Temporal Predicates and Their Visual Specification

*Abstract.* Temporal changes of spatial objects induce modifications of their mutual topological relationships over time. Topological relationships describe the relative positions of spatial objects to each other (like *disjoint*, *overlap*, *meet*, *inside*) and are independent of metric aspects. *Spatio-temporal predicates* describe developments of these well-known relationships. For example, a plane can be first disjoint from a hurricane, then meet its boundary, be inside of it for some time, meet its boundary again, and then reach the outside of the hurricane. In other words, the plane crosses the hurricane. A formal concept has already been designed for spatio-temporal predicates, which can be used in queries on spatio-temporal databases. The remaining, important, and challenging task is to find an efficient and numerically robust implementation for it. The unlimited set of possible spatio-temporal predicates that can be specified by the user has lead to the proposal of a visual interface. This interface supports the specification on an intuitive and easily manageable level and leads to the design of a visual language. The task here is to implement such a user-friendly visual user interface. Another important aspect is to design mechanisms to integrate these types into an extensible database system.

*Remark:* This topic is based and dependent on Topic 3.

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* databases in general, spatio-temporal databases, spatial databases, computational geometry; programming skills in C++

*Introducing Literature:*

Martin Erwig & Markus Schneider. Spatio-Temporal Predicates. *IEEE Transactions on Knowledge and Data Engineering* (*TKDE*), 2002, accepted for publication.

Martin Erwig & Markus Schneider. Query-By-Trace: Visual Predicate Specification in Spatio-Temporal Databases. *5th IFIP 2.6 Working Conf. on Visual Database Systems* (*VDB*), 2000.

Martin Erwig & Markus Schneider. Developments in Spatio-Temporal Query Languages. *IEEE Int. Workshop on Spatio-Temporal Data Models and Languages* (*STDML*), 441-449, 1999.

Martin Erwig, Markus Schneider, Ralf Hartmut Güting & Michalis Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica*, 3(3), 265-291, 1999.

Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos Markus Schneider & Michalis Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems* (*TODS*), 25(1), 1-42, 2000.

Luca Forlizzi, Ralf Hartmut Güting, Enrico Nardelli & Markus Schneider. A Data Model and Data Structures for Moving Objects Databases. *ACM SIGMOD Int. Conf. on Management of Data* (*SIGMOD*), 319-330, 2000.

### *Topic 6:* Conceptual Modeling of Database Applications Including Spatial, Temporal, and Spatio-Temporal Aspects

*Abstract.* A conceptually "clean" design should be the prerequisite of all database applications. The conceptual level as one of the three abstraction levels of the architecture of a database system has the goal to describe and structure the application area of interest (for example, a large company). This level is independent of the database system used. Two frequently deployed tools for performing the conceptual design are the *Entity-Relationship Model* (*ER Model*, *ERM*) and the *Unified Modeling Language* (*UML*). They are well suitable for standard alphanumerical data and corresponding applications. The interesting issue is now to which degree the kind of data (for example, alphanumerical, video, multimedia, spatial data) has influence on the conceptual design method. The task here is extend the ERM and UML by spatial, temporal, and spatio-temporal concepts in an elegant way. The approach has then to be realized in an extended ERM and/or UML editor, or integrated ERM/UML editor. This editor should allow an automatic transformation of a conceptual schema to a database schema (for example, relational, object-relational, object-oriented schema).

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* ERM, UML, conceptual design, databases in general, spatial databases; programming skills in C++

*Introducing Literature:*

Peter Chen. The Entity-Relationship Model: Towards a Unified View of Data. *ACM Transactions on Databases Systems* (*TODS*), 1 (1), 1976.

Object Management Group. *UML Resource Page*. http://www.omg.org/uml/, 1997-2002.

*Topic 7:* The Integration of Spatial Applications on PDAs

*Abstract.* PDAs are nowadays affordable handhelds for mobile and wireless computing. Due to their limited size, PDAs are restricted with respect to main memory and CPU power. This topic investigates the integration of spatial applications on PDAs which require special and usually extensive resources. These resources include graphical visualization of spatial objects, both raster- and vector-oriented, GPS (Global Positioning System) for locating the current position in space, and the existence of a 'spatial database system' for storing, retrieving, querying, and manipulating geometric (and thematic) data. The issues here relate to the needed features of such a spatial database system and to the realization of data structures and algorithms under these restricted preconditions. A query language provided to the user should certainly not be SQL, which is the database standard and has a certain complexity, but a simpler, voice- and command-oriented one. Another important aspect is how to download data from an external resource into the PDA. A detailed investigation of what is going on in the market is indispensable.

*Appropriate for:* Master

*Status:* assigned

*Desired or required knowledge:* PDAs, mobile computing, databases in general, spatial databases, computational geometry; programming skills in C++ and/or Java

*Introducing Literature:*

none (It is difficult to find good references. Most literature of interest is commerially-oriented.)

*Topic 8:* Design and Implementation of a Type System for Two-Dimensional Crisp and Vague Spatial Data by Using the Functional Programming Language Haskell

*Abstract.* This topic deals with the design and implementation of *two-dimensional spatial data types* like *point*, *line*, and *region* in the Euclidean plane by using the functional programming language *Haskell* as implementation language. A functional programming language gives a simple but powerful model of programming, namely functions: one value, the result, is computed on the basis of other values, the inputs. It provides central concepts in modern programming, including abstraction (in a function), data abstraction (in an abstract data type), genericity, polymorphism and overloading. The task is to investigate to which degree Haskell is appropriate to implement such a type system and to compare it with traditional imperative or object-oriented programming.

Although some spatial data type extension packages have already been implemented in the past (in an imperative or object-oriented manner), these still suffer from weaknesses like lacking generality, missing functionality, and non-existing numerical robustness. The spatial type system to be constructed is to include crisp and vague spatial data. *Crisp* spatial objects have a precisely determined boundary like countries and districts with their political and administrative boundaries or land parcels with their cadastral boundaries. *Vague* spatial objects do either not have sharp boundaries, or their boundaries cannot be precisely determined. Examples are land features such as population density, soil quality or vegetation, oceans, biotopes, deserts, an English speaking area, clouds, and sandbanks. For testing and demonstrating purposes, a graphical user interface for the visualization of 2D spatial objects and the evaluation of operations should be included. Another interesting and important aspect is to design mechanisms to integrate this type system into an extensible database system.

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* spatial databases, computational geometry; programming skills in the functional programming language Haskell

*Introducing Literature:*

Ralf Hartmut Güting. An Introduction to Spatial Database Systems. *VLDB Journal* (*Special Issue on Spatial Database Systems*), 7(3), 231-246, 1994.

Markus Schneider. *Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems*. LNCS 1288, Springer-Verlag, 1997.

Martin Erwig & Markus Schneider. Vague Regions. *5th Int. Symp. on Advances in Spatial Databases* (*SSD*), 298-320, 1997.

F.P. Preparata & M.I. Shamos. *Computational Geometry.* Springer Verlag, 1985.

Simon Thompson. *Haskell: The Craft of Functional Programming*. Addison Wesley, 1999.

Jörn Dinkla. *Library for Geometric Algorithms in Haskell*. http://www.dinkla.net/fp/cglib.html, 2001

*Topic 9:* Metric Refinements of Topological Relationships

*Abstract.* Topological relationships (like *overlap*, *disjoint*, *meet*) between spatial objects (like *regions*, *lines*, *points*) have been a focus of research on spatial data handling and reasoning for a long time. Especially as predicates they support the design of suitable query languages for spatial data retrieval and analysis in databases. Whereas research on this topic has always been determined by *qualitative* methods and by an emphasis of a strict separation of topological and metric, i.e., *quantitative*, properties, the task here is to investigate their possible coexistence and cooperation. Metric details can be exploited to refine topological relationships and to make important semantic distinctions. The metric refinements introduced in this paper have the nice feature of being topologically invariant under affine transformations. Since the combination of a topological predicate with a metric refinement leads to a single unified quantitative measure, this measure has to be interpreted and mapped to a lexical item. This leads to fuzzy topological predicates, and we demonstrate how these predicates can be integrated into a spatial query language.

*Appropriate for:* Master, Individual Study

*Status:* available

*Desired or required knowledge:* spatial databases, topological relationships, computational geometry; programming skills in C++/Java

*Introducing Literature:*

Ralf Hartmut Güting. An Introduction to Spatial Database Systems. *VLDB Journal* (*Special Issue on Spatial Database Systems*), 7(3), 231-246, 1994.

Markus Schneider. *Spatial Data Types for Database Systems – Finite Resolution Geometry for Geographic Information Systems*. LNCS 1288, Springer-Verlag, 1997.

F.P. Preparata & M.I. Shamos. *Computational Geometry.* Springer Verlag, 1985.

Max J. Egenhofer, Andrew Frank, and J. P. Jackson. A Topological Data Model for Spatial Databases. *1st Int. Symp. on the Design and Implementation of Large Spatial Databases*, LNCS 409, pp. 271-286. Springer-Verlag, 1989.

Max J. Egenhofer and John Herring. A Mathematical Framework for the Definition of Topological Relationships. *4th Int. Symp. on Spatial Data Handling*, pp. 803-813, 1990.

Max. J. Egenhofer and R. D. Franzosa. Point-Set Topological Spatial Relations. *Int. Journal of Geographical Information Systems*, 5(2):161-174, 1991.

Eliseo Clementini, Paolino Di Felice, and Peter van Oosterom. A Small Set of Formal Topological Relationships Suitable for End-User Interaction. *3rd Int. Symp. on Advances in Spatial Databases*, LNCS 692, pp. 277-295, 1993.

Max J. Egenhofer and A. R. Shariff. Metric Details for Natural-Language Spatial Relations. *ACM Transactions on Information Systems*, 16(4):295-321, 1998.

Thomas Behr & Markus Schneider. Topological Relationships of Complex Points and Complex Regions. *Int. Conf. on Conceptual Modeling*, pp. 56-69, 2001.

Markus Schneider. A Design of Topological Predicates for Complex Crisp and Fuzzy Regions. *Int. Conf. on Conceptual Modeling*, pp. 103-116, 2001.

### *Topic 10:* A C++-Library for the Postscript Programming Language

*Abstract.* Postscript is a well known and frequently used page description or page layout programming language. It is often used for producing print-outs on laser printers. The task is to study the manifold functionality of Postscript and to provide it in a C++-library.

*Appropriate for:* Individual Study

*Status:* available

*Desired or required knowledge:* Postscript as a programming language; programming skills in C++

*Introducing Literature:*

Adobe Systems Incorporated. *PostScript(R) Language Reference*. Addison-Wesley, 1999.

Henry McGilton & Mary Campione. *PostScript(R) by Example*. Addison-Wesley, 1992.

*Topic 11:* Exploiting Data Compression Techniques for Processing and Querying in Database Systems

*Abstract.* Usually data is stored, processed, and queried in an uncompressed format in a database. This also holds for commercially available database systems. The first task of this thesis is to study whether and how proposals have been made in a database context to exploit data compression. The second task then is to find out how existing data compression techniques can be used or new ones can be designed to advance database technology. It is of particular interest how these techniques can be used for accelerating query processing. Another application is bulk loading of data. All these techniques are in general of great interest for non-standard applications with very large data volumes like video databases, multimedia databases, spatial databases, spatio-temporal databases, and biological databases.

*Appropriate for:* Individual Study

*Status:* assigned

*Desired or required knowledge:* databases in general, data compression techniques; programming skills in C++

*Introducing Literature:*

(none, has to be searched for)

*Topic 12:* Redesign and Reimplementation of the CISE Department's Web Site

*Abstract.* This topic relates to the redesign and reimplemention of the web site of the CISE department. Several subtasks can be identified: (i) Determine the technology that is employed for the current web site, (ii) study and compare a number of (computer science-related) web sites and learn from their design, (iii) study the literature on web design, (iv) make a survey/questionnaire to learn about the opinions, thoughts, and suggestions of the faculty members, the staff, and the students, (v) formulate a catalog of requirements (e.g., searching machine), (vi) design a concept for the web site, (vii) let the concept be evaluated by the users, and (viii) implement the concept.

*Appropriate for:* Master

*Status:* available

*Desired or required knowledge:* Web design, HTML, related technologies

*Introducing Literature:*

(none, widespread literature that has to be searched for)