Transactions in GIS, 2000, 4(1): 23–42

Research Article

Modelling the Real World: Conceptual Modelling in Spatiotemporal Information System Design

Agnar Renolen
Norwegian University of Science and Technology
Norway

Abstract
Throughout the relatively young history of research on spatiotemporal modelling, a substantial number of models have been presented. However, since a spatiotemporal model represents a closer approximation to the real world than is the case for traditional models, such models must be based on a thorough understanding of how objects 'behave' in reality. One way to acquire such knowledge is using *conceptual modelling methodologies*. In this paper, an overview of different modelling principles and a selection of *conceptual modelling languages* are presented together with examples related to a selection of spatiotemporal problems.

1 Introduction

Recent research has identified the need to handle historical information in geographical information systems (GIS). The various aspects of temporal GIS and spatiotemporal models have therefore been an active research field since the late 1980s, see Al-Taha et al (1994), Langran (1992) and the references therein.

Historically, research in GIS has focused on application issues of digital cartography such as how to represent and manipulate spatial data structures in computers. The traditional representation schemes for geographical information systems have utilized cartographic primitives such as points, lines, and areas. However, modern computers are capable of representing more information and knowledge about the real world than the paper map model is able to convey, such as the temporal perspective of spatial information.

Address for correspondence: Agnar Renolen, Department of Surveying and Mapping, Norwegian University of Science and Technology, Trondheim, Norway. Email: agnar.renolen@telenor.com

In order to acquire and communicate phenomena in the real world, it is necessary to describe these phenomena at the *conceptual level*. Within the field of information systems engineering, an abundance of conceptual modelling languages to describe various aspects of the real world have existed for a long time. But 'Geographical information systems are often built without due considerations to this discipline...' (Hadzilacos and Tryfona 1996).

Wand and Weber (1989) provide the following definition of an information system:

> *An* information system *is a human-created representation of a real world system as perceived by somebody, built to deal with information processing functions in organizations*.

A *spatiotemporal information system* (STIS) is defined here as an information system where the spatial location and temporal history of the real world system are of interest to the organizations.

Early research on spatiotemporal models for GIS has focused on the development of computer models that are based on simplified concepts such as those that only timestamp records. These so-called *change-based* approaches have drawbacks, such as the lack of ability to model continuously changing objects (Al-Taha and Barrera 1990). One example of a change-based model where spatiotemporal objects have spatial and (bi-)temporal extent has been described by Worboys (1994a, 1994b).

However, a better understanding of these aspects is required to create models that to a larger extent capture the semantics of the real world. In the literature, we have seen important work on creating conceptual frameworks for modelling spatiotemporal phenomena (Langran and Chrisman 1988, Peuquet 1994, Claramunt and Thériault 1995, Spaccapietra et al 1998). This article describes work done on the modelling of various temporal (and spatial) aspects of a real world system, in a sequel referred to as the *universe of discourse* (UoD), using standard and specialized conceptual modelling languages already described in the literature.

The remainder of this paper is organized as follows. The next section briefly reviews some issues of conceptual modelling, and the following sections discuss the most important models with respect to the design of an STIS. The paper closes with some concluding remarks.

2 An Introduction to Conceptual Modelling

Computers and computer languages are generally abstract. Larger systems are compiled from thousands of statements written in source code and it is virtually impossible for even a skilled programmer to get a general view of a large program without any visual support in the form of figures and diagrams. During the development of computer systems it is important that all participants understand the problem domain. In general, software designers and software users represent different levels of knowledge about programming, and communication between them may easily be distorted due to misunderstandings and lack of insight. The key to successful communication among participants involves sharing relevant conceptual knowledge about the domain of discourse. This can usually be achieved by developing so-called *conceptual models* (Sølvberg and Kung 1993).

A conceptual model is usually presented in diagrammatic form ('boxology') with a grammar that consists of boxes and links between them. Sølvberg and Krogstie (1996) provide the following definition of a conceptual model:

> A conceptual model *is the phenomenon of a domain at some level of approximation externalized in a semi-formal or formal language.*

In information systems engineering, a conceptual model serves as a tool for sense-making, as a vehicle for communication, and as the documentation and basis for design and implementation. However, conceptual modelling techniques are not only useful in the design and development of computer data structures, they have also proven to be a valuable methodology in the acquisition of knowledge of real world phenomena.

Interestingly, cartographic maps and conceptual models have much in common. A map can be defined as a selective, symbolic, generalized image of an object presented at a given scale (Bjørke 1995). Comparing this definition of a map with the definition of a conceptual model given above, one could conclude that a map is a conceptual model. A map represents a phenomenon of a domain (a part of the real world seen from above), it represents some level of approximation (it is selective and generalized), and it is presented in a formal language (a symbolic image where the symbols are defined through some legend). This may explain why the paper map model has been such a popular basis for spatial computer models in GIS.

2.1 A Brief Review of Conceptual Modelling Perspectives

Throughout the history of computers and computer systems development, an abundance of different conceptual modelling languages has been presented. Two well-known modelling languages utilize entity-relationship diagrams and data-flow diagrams. In general, modelling languages can be divided into classes according to the *structural principle* or the *perspective* of the language. In Sølvberg and Krogstie (1996), the following seven perspectives are described:

The structural perspective: The focus of the structural perspective is on data and data modelling. The main components of structural models are entities, relationships, attributes and constraints on relationships (i.e. cardinalities). It was the development of the *entity-relationship* language of Chen (1976) that represented the breakthrough of this modelling perspective.

The functional perspective: The functional perspective focuses on *processes* rather than on objects and physical entities. The best known conceptual modelling language with a process perspective is the *data flow diagram* (DFD) which describes the UoD in terms of external entities, processes, data stores and (data-) flows between these.

The behavioural perspective: The basic concepts of the behavioural perspective are *states* and *transitions* that transform the system from one state to another. One of the problems with this perspective is that larger and complex systems quickly become unmanageable with an almost infinitely large number of possible states. To overcome this, some languages such as Statecharts add hierarchical abstraction mechanisms in the form of AND and XOR decompositions (Harel 1987).

The rule perspective: The main application of the rule perspective is in knowledge systems and artificial intelligence (AI). In general, a rule has the form:

if<condition> **then**<consequence>

Rules are both utilized to describe knowledge about the real world (e.g. in knowledge databases) and to express constraints on other conceptual models (e.g. on an ER-model). One drawback with rules is that conditions are expected to be either true or false, while in cartography as in many other applications, natural conditions often seem to have a fuzzy nature.

The object-oriented perspective: The object perspective has basically emerged as a result of the need to support object-oriented programming languages like SmallTalk, C++ and Eiffel. However, object-oriented analysis and design have truly become a branch on its own, applying the same concepts that were introduced in the object-oriented programming languages (i.e. the object-oriented paradigm).

The communication perspective: The communication perspective is based on the assumption of language/action theory developed by Austin and Searle called the *speech act theory* (Austin 1962; Searle 1968, 1979). A few modelling languages exist such as *action workflow* diagrams (Medina-Mora et al 1992).

The actor role perspective: The actor and role perspective is based on ideas developed during work on object-oriented programming languages and intelligent agents in AI. Basic constructs of this perspective are actors, roles and agents.

At first glance, it is quite evident that several of these perspectives are useful in the development of an STIS. Behavioural models may help us understand how objects change over time and functional models may possibly support us in that process. Structural models on the other hand normally represent the real world in a *static* fashion, although extensions to the ER-language that support changes over time, such as the ERT-model (Theodoulidis et al 1991, McBrien et al 1992) have been developed. The object-oriented approach is closely related to the structural approach since it describes relationships between objects, but since the approach incorporates the functionality of the objects, it is also possible to draw relations to the functional approaches. Moreover, object-oriented models may further be supported by *Objectcharts* (Coleman et al 1992) which is an object-oriented adaption of Statecharts (Harel 1987).

The rule perspective is also of great value in the development of temporal information systems in general. Rules that are coupled with business policy may change over time, and historical data should be viewed in context with the current policy at the time in question. The ability to express policy in the form of explicit rules is therefore critical.

2.2 Information, Data, and Model Domains

The terms *data* and *information* are often used interchangeably in the literature. However, a clear distinction between these two concepts should be maintained. *Data* should be considered as a collection of symbols represented in computer-readable form. Data exists in the form of bits/bytes in computer files, whereas *information* involves some kind of human interpretation. For example, a coastline may be represented as a stream of points connected with straight line segments. This piece of data conveys some information that a user may interpret as fjords, bays and peninsulas, when presented in a graphical format. In other words, information is associated with a higher perception level, whereas data is associated with low level computer representation.

Along this line of perception levels, three domain levels of information system modelling have been identified (Sølvberg and Kung 1993). These are:

The subject domain: Concerns itself with information about the real world. Focus is on physical entities or abstract concepts such as persons, parcels, roads or legislation.

The interaction domain: Concerns itself with the way information in the information system is to be presented and perceived by the system users.

The implementation domain: Concerns itself with the low level implementation of information systems. Focus is on data, communication protocols, data access and algorithms.

A top-down approach to information systems modelling begins with the analysis phase and the study of the real world with the aim of creating a subject domain model. Then the analysis phase moves on with the user interaction modelling, and ends up with the design of the implemented system. In the GIS literature, we have seen the opposite, i.e. a bottom-up approach. First the data model has been designed, and then the user has to fit the real world into the confines of this model. The same trend can be seen in the suggested spatiotemporal data modes such as the space-time composite vector model (Langran and Chrisman 1988) or the event oriented spatiotemporal model (Peuquet and Duan 1995). In general terms, the focus has been on the data rather than on the information.

3 Structural Modelling and the Time Dimension

3.1 Entity Relationship Models

The Entity Relationship model, or in short the ER-model, developed by Chen (1976) was not the first language for semantic data modelling, but it certainly became the most popular. The main reasons for this popularity can be traced to the simple diagrammatic representation and easy transition to tables of relational databases. Although the intention of the ER-modelling language was to describe the structure of (relational) databases, the ER-language is also appropriate for modelling general knowledge about the real world (i.e. not only the part to be stored in the database).

There are two basic constructs of the ER-model: *entities* and *relationships*. Chen defined an entity as 'a thing that can be distinctly identified' and a relationship as 'an association among entities'. This wide definition is a strength of the ER-model. Each entity is characterized by a set of *attributes* which is common to all entities of the same type.

Consider a temporal database of countries and cities of the world. In this model countries are bounded by borders and coastlines. A coastline may also bound small islands, hence a country may be associated to more than one coastline whereas one coastline may bound several countries. Borders separate two countries, and each country has a capital and a number of other cities. Figure 1 shows an ER-diagram of this schema.

However, for modelling concepts in the real world, the ER-language has some shortcomings. One problem is the lack of support of attributes in the original language. It is therefore common to see the EAR (entity-attribute-relationship) model in the literature. Other extensions, such as the EER (extended entity-relationship) model also

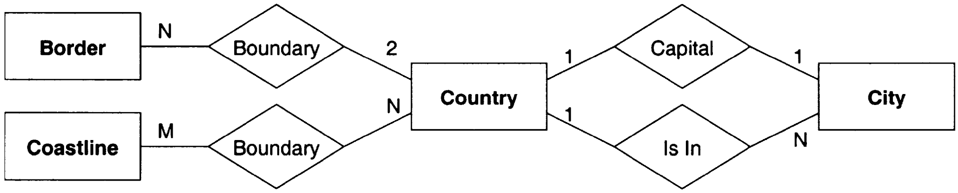

Figure 1 A sample ER model describing cities and countries of the world.

include concepts such as sub-typing and association; concepts that today are known because of the object-oriented model (Teorey et al 1986).

3.2 The ERT Language

A natural consequence of the increasing research on temporal databases is the emergence of accompanying conceptual modelling languages. Since much of the research on temporal databases has focused on extensions to the relational model, it is not surprising that most conceptual modelling languages supporting time are extensions to the ER or the EER-model. One such language is the *ERT* (Entity-Relationship with Time) modelling language (Theodoulidis et al 1991, McBrien et al 1992). In this language, which is based on the EER-model, support for temporal concepts has been applied by the use of *time-stamping* of entities and relationships.

The basic constructs of this language are the *entity class* (denoted by a rectangle) which denotes a set of objects which share the same set of attributes, the *value class* (denoted by a rectangle with a black corner) which is used to describe an entity's attributes, and the *relationship* (denoted by a line with a small black square) which describes the associations between an entity and a value class or another entity. Furthermore, the inheritance extension is represented by a relationship with a circular join. If the circle is solid, then the sub-classes are disjoint or total; if open, then the sub-classes are overlapping or partial.

To implement the temporal dimension in the ERT diagrams, entity classes and relationships may be either **T**- or **H**-marked. If an entity class is **T**-marked it means that the entity only exists at certain times (or ticks) in our UoD, meaning that the entity is undergoing *temporal variation*. If a relationship is **T**-marked, it means that the relationship between the two entities exists for only a subset of the time (number of ticks) for which both of the entities exist. If a relationship exists between two entities of which at least one is **T**-marked and the other is not **T**-marked, it means that the relationship exists as long as both entities co-exist in our UoD.

The **H**-mark is used to indicate that a relationship has an *historical perspective*. This means that a relationship may exist between two entities that do not co-exist in time. For example, we may say that one *person* has a grandparent that is another *person*, but the two persons did not co-exist in our UoD if the grandparent died before the grandchild was born. However, in the grandparent example, we might want to say that the grandparent is related to its grandchild from the time that the grandchild begins to exist. For this purpose, the **TH**-mark may be used.

Figure 2 shows an improved model of the temporal map over Europe using the ERT-language. In this model, we have distinguished two types of countries: monarchies and republics. For simplicity, we assume that all countries are either

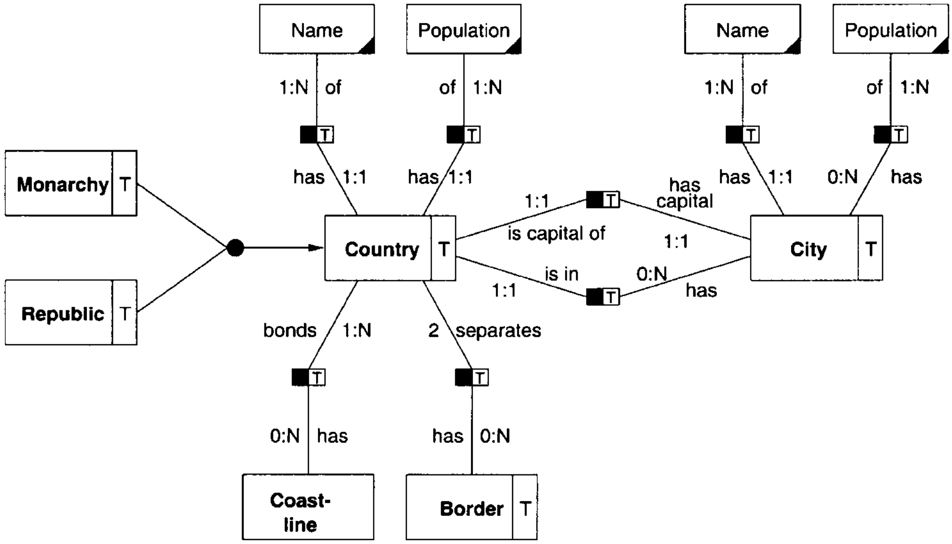

Figure 2 An example of an ERT model showing a temporal map over Europe.

monarchies or republics, hence the generalization link is a solid circle. Furthermore, we have also added names and populations of cities and countries as attributes and added the T-marks wherever appropriate. Because the ERT model both supports temporal aspects and sub-typing it is particularly interesting in the design of an object-oriented system. Furthermore, the notation of the ERT-language allows us to read out the relationships, as for example the relationship between countries and their capitals: A country *has* one capital, while a city can be the *capital of* one country.

4 Modelling Processes

There are several reasons why it is desirable to model processes in the real world. One is because they involve human interaction, and these processes need to be automated (e.g. monitoring and management of ships in a harbour). Other opportunities may arise because the process is to be simulated in the computer (e.g. the melting and accumulation of glaciers). This section presents two modelling languages that belong to the functional perspective, viz. *data flow diagrams* and *demos activity diagrams*.

4.1 Data Flow Diagrams

There are two commonly used notations for data flow models. One was proposed by DeMarco (1978) and the other was proposed by Gane and Sarson (1978). The two languages contain exactly the same concepts, but provide different symbols for them. For practical reasons, we use the notation of Gane/Sarson, which is shown in Figure 3. Data flow diagrams have many applications in STIS. In contrast to the ER-diagrams they do not show how things are, but how things are done or how things happen. This is valuable information in STIS.

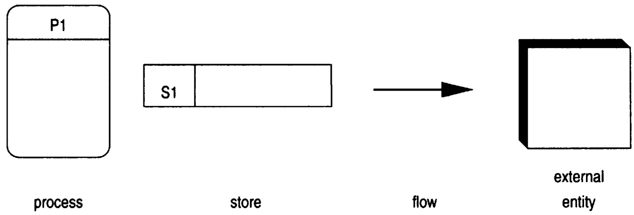

Figure 3 The functional perspective: Symbols of the DFD language.

Consider a building department that maintains a database for all buildings in a municipality. In order to set up new buildings, pull down or renovate existing buildings, or to change the use of buildings, the proprietor has to apply and get permission from the department to do so. The application is received and verified for conformance with the area development plan, and notifications are sent out to the neighbours for reactions. Upon approval (or rejection) of the application, the necessary documents are sent back to the proprietor. When a new building is built or an existing one is changed, it is inspected to verify that it conforms with the original application and the final position of the house is surveyed. In some cases, houses get damaged (partly or totally) due to fire or other natural causes (e.g. landslides and avalanches). Notification about this is received from the insurance companies. This way, the building department will have a complete inventory of all buildings in their municipality at all times.

Figure 4 shows a data flow diagram of the system described above. In this diagram the external entities are the proprietors, their neighbours and the insurance companies. The main data store is the building database, but also a cadastral database and area development plan are needed. The latter two databases may be maintained by other departments. The processes shown in the diagram are described as follows:

P1: Receive application: An application is received from a proprietor. The application may be either for setting up a new building, to extend or restore an existing one, or to change the use of a building (e.g. to change from private residence to an office building). The application is checked whether all necessary documentation has been included, and if it is okay, it is forwarded for processing.

P2: Handle application: In this process, the neighbours to the building site are determined from the cadastral database, and a notification is sent out to them. If the neighbours have objections, they will have a deadline to submit them. If there are no objections and the application is in conformance with the area development plan, it is accepted, and forwarded for final notification to the proprietor. If a new building or a wing is to be set up, preliminary coordinates for the position of the house are added to the building database.

P3: Answer application: The application is accepted, and the necessary certificates and permissions are issued to the proprietor.

P4: Final control: When the building is finished, it is inspected in order to verify that the building, restoration or change in use matches the application and the original intentions.

P5: Survey Building: If a new building or wing has been set up, the new building is surveyed and its exact location is determined. The results are stored in the building database.

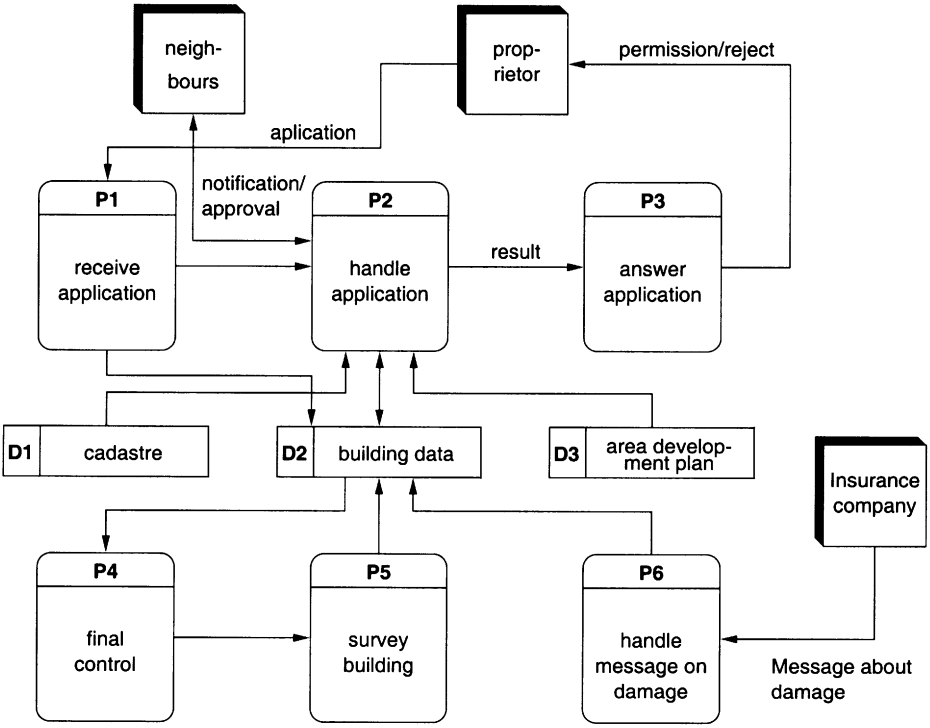

Figure 4 A data flow diagram of the activities in a municipal building department.

P6: Handle message on damage: Sometimes, houses are damaged, partly or totally, due to fire, landslide, avalanche, or other reasons. Notification about this is received from the respective insurance companies, and the building database is updated accordingly.

In the diagram, it is easy to see which processes contribute to the updating of the databases, and which processes only need to read information from databases. By further specification, it is also possible to describe exactly what information is updated by these processes.

4.2 Demos Activity Diagrams

Another application of the functional perspective is where real world processes are to be simulated in computers. In fact, computer simulation is itself an active research field, and designers of STIS may learn from this field. An early simulation language, Simula (Dahl and Nygaard 1966), has been extended with a package called Demos (Birtwhistle 1979) to ease the implementation of simulation programs. Demos programs can be visualized using so-called *Demos activity diagrams*, and the original notation has been extended a number of times by Birtwhistle (1979) and Hughes and Rønningen (1985).

An activity diagram shows how an object enters the UoD, goes through different activities, acquiring resources, cooperates with other objects, interrupts other activities or is interrupted by other activities, before it leaves the UoD. In addition to activities,

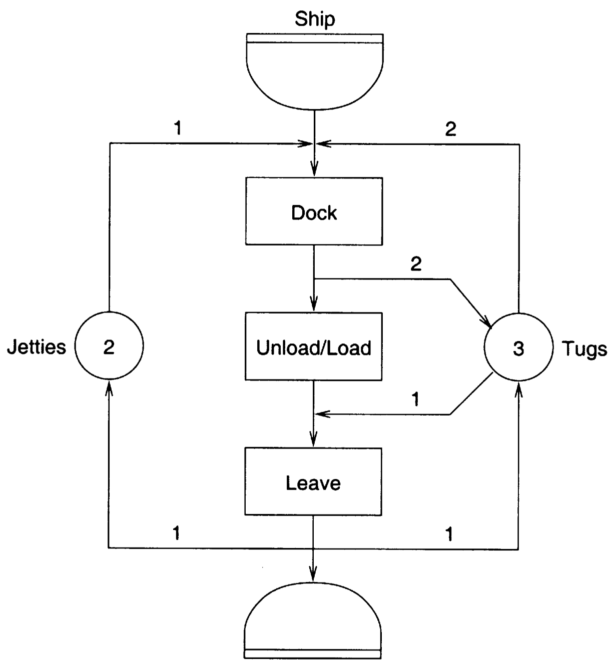

Figure 5 Demos activity diagram for a harbour.

activity diagrams also incorporate entities such as resource objects, bin objects, conditionals and wait activities.

Figure 5 shows a sample activity diagram for a harbour that is managed by the harbour administration with the help of an STIS. The harbour administration has two jetties and three tug boats which are modelled as resources (indicated by circles). Two tugs are required to dock a ship whereas one is sufficient when a ship is leaving. When a ship arrives, i.e. enters the UoD (indicated by a lower half circle) it must acquire two tugs and one jetty. If these resources are not available, the ships have to wait until they become available, and they are served on a first come, first served basis. The processes of the system, such as the docking process, unloading/loading process and the undocking process are indicated by rectangles, and these processes are considered to take some arbitrary amount of time. When the docking is complete, the two tugs are released, and the unloading and loading process can begin. When the loading is complete, one tug is acquired before the ship can leave. The ship is towed to sea, the jetty and tug are released, and the ship can leave the UoD (indicated by a upper half circle).

The above example only illustrates a subset of the concepts that can be incorporated in demos activity diagrams. Nevertheless, demos activity diagrams share several common features with the behavioural perspective; the example above could to some extent also be modelled using Petri-nets.

5 Modelling Behaviour

Understanding behaviour is one of the most fundamental issues of STIS engineering. Most spatiotemporal models so far are extensions of existing data models, and a

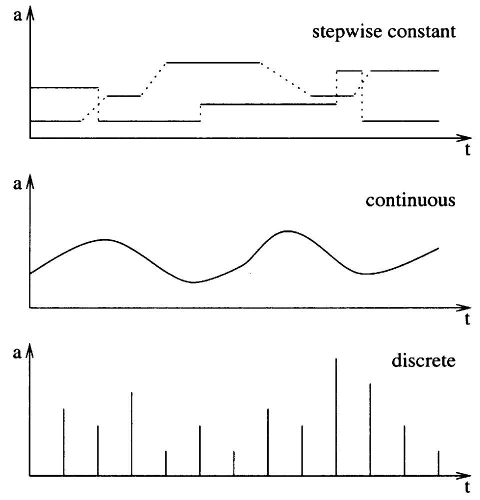

Figure 6 Temporal behaviour of an attribute.

common solution is to simply timestamp the data when it is updated. Such systems are only capable of representing changes as sudden events. However, we know that many changes in the real world have duration. In general, features in the real world exhibit a wide range of temporal behaviour, and three basic types of behaviour have been identified (Segev and Shoshani 1993, Montari and Pernici 1993) as illustrated in Figure 6:

Stepwise constant: A feature of this type is considered to be static and changed by events. These events may be instantaneous, such as the division of a parcel, or they may have duration such as the building of a road, or the change in position of a ship that is sailing from one harbour to another.

Continuously changing: Features of this type are always considered to be in a changing state. The population of the country or city, or the expansion and retreat of glaciers are examples of such behaviour.

Discrete values: Features of this type are considered to be associated with specific times or time intervals. The amount of precipitation per day, and the gross domestic product of a country are examples of such behaviour.

To model temporal behaviour, it seems natural to use a language such as Statecharts (Harel 1987) or Objectcharts (Coleman et al 1992). However, these languages are similar to finite state machines which are based on the idea that the system is always in one state and that transitions between each state are instantaneous. Although, many objects in the real world exhibit such behaviour, it would be of great value if we could model gradually changing objects as well. However, if we introduce the *state of change* as a distinct state, we may obtain a generic Statechart model (i.e. a meta-model) for spatiotemporal objects as shown in Figure 7. According to this model, an object is either *alive* or it is *dead*. An object that is alive, may die and enter the state of being *dead* and later become alive again by a *reincarnation* event. If the object is alive, it may either be in a static state, or in a state of continuous change. An object that is in a changing state may *stabilize* and enter the state of being static, and then later it may start to change again. An object that is in a static state may change instantaneously and continue to be in the static state.

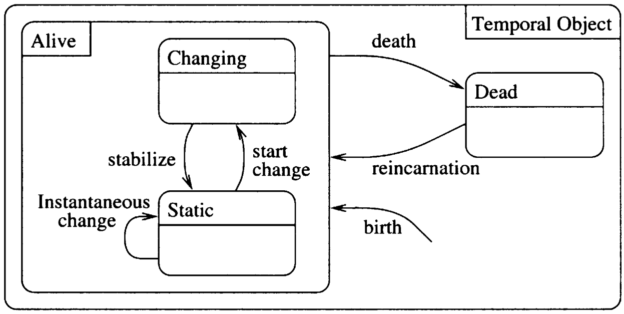

Figure 7 Generic behaviour of temporal objects using the Statechart notation.

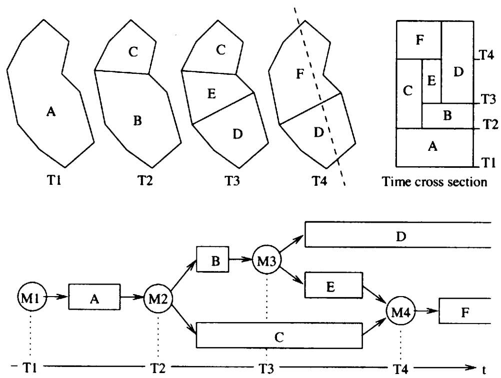

Figure 8 The story of a land area (above) shown in the history graph notation (below).

It was the idea of this model that led to the definition of the *history graph* notation (Renolen 1996). In this language, which in some way is similar to Petri-nets (which is another modelling language that belongs to the behavioural perspective; Petri 1962), the objects of a data set may be described through a series of consecutive states (i.e. static states) and changes (i.e. changing states). The states are denoted by a square rectangle, while the changes are denoted as boxes with circular ends. Both the states and changes are associated with a time interval, and the boxes are stretched to mimic the time interval they are associated with. Objects that change suddenly would then be described by transitions with zero duration (i.e. events), while objects that change continuously would be described by version with zero duration (i.e. snapshots) describing intermediate states. An object that is dead, is denoted with a rectangular box with a dashed outline. Figure 8 shows a sample story where a region is split and merged. Since the changes in this story are considered as sudden events, the transitions are shown as circles. A similar language has been proposed by Hornsby and Egenhofer (1997), but it does not consider changes as distinct entities.

Studying Figure 8, one can identify a number of distinct change types, such as the splitting and merging of objects. As illustrated in Figure 9, a total of seven different types of changes were identified. These are as follows (a refinement of these change types can be found in Claramunt and Thériault (1995) and Hornsby and Egenhofer (1997)):

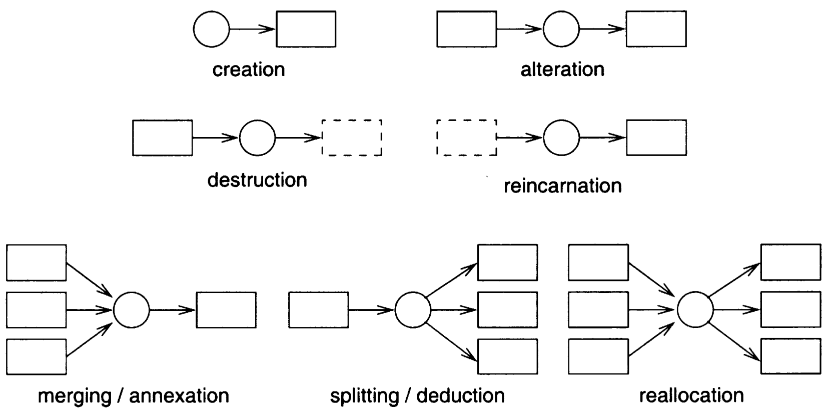

Figure 9 Seven basic types of changes (shown in history graph notation).

Creation: An object is created.
Alteration: An object is changed or modified.
Destruction: An object is destroyed or removed.
Reincarnation: An object that previously has been destroyed or removed is reintroduced, possibly with a new state and location.
Split/Deduction: An object is subdivided into two or more new objects or one or more objects is deducted from an existing object.
Merge/Annexation: Two or more objects are joined together to form a new object or one or more objects are 'swallowed' into another object.
Reallocation: Two or more objects are merged together and two or more different objects result from the change.

Although the history graph notation helps us understand temporal behaviour in particular cases, it does not allow us to describe the general behaviour of certain types of features. For example, changes to a country can be one of the following types:

1 Two countries may merge to form a new country, e.g. East and West Germany merge to form Germany.
2 One country splits to form two countries, e.g. Czechoslovakia splits to form the Czech Republic and Slovakia.
3 A region in one country gains its independence and forms a new country, e.g. Estonia, Latvia and Lithuania withdraw from the Soviet Union.
4 A country is annexed into another country, e.g. Iraq invades Kuwait.
5 A border between two countries is adjusted according to an agreement between the two countries.
6 The population is changed. Apart from the continuous change of population by natural birth and mortality, all of the events above will have an effect on the population of the involved countries.
7 When changes occur in territories, some cities will change country, and the affected countries will experience an increase or decrease in population.
8 The capital of a country is moved to another city, e.g. Germany moves the capital from Bonn to Berlin.
9 A country changes name, e.g. the Soviet Union becomes Russia.

On the basis of these changes, one can make transition specifications. Formally in a Statechart, a transition specification comprises the initial and the final state of the transition and the service name for the transition, together with a precondition and a postcondition.

A problem with the Statechart is that it poorly expresses the interaction between objects, e.g. how can we comprehensively describe a deduction. A deduction generally means the creation of one object and alteration of another. Obviously, such issues have not been addressed, although the history graph notation to a large extent can assist in such specifications.

6 Object-Oriented Models

A popular approach in GIS modelling in general but also in spatiotemporal modelling is the *object-oriented* approach, and a number of data models have been presented (Ramachandran et al 1994, Worboys 1994, Hamre 1995). However, the object-oriented model has been used in GIS development as a mere wrapper around the vector model, rather than a fundamental approach according to which geographical knowledge could be modelled. Apart from the advantages of the object-oriented model in traditional GIS, such as increased modelling power, Käfer et al (1994) list four main advantages of an object-oriented model in a temporal database:

1 The complete history of an entity can be encapsulated into one single object.
2 Since the complete history of an entity can be represented as a single object, queries become less complicated, because they do not consider the dispersion of the entity over many tuples.
3 Since complex object queries are executed efficiently, the corresponding temporal data should be handled efficiently as well.
4 Handling of temporal and non-temporal data can be accomplished in a uniform way.

The object-oriented approach provides concepts such as object class, association, aggregation, and generalization. Several object-oriented modelling languages exist (Edwards and Henderson-Sellers 1993, Martin 1993, Booch 1996), but the Object Modelling Technique (OMT) language (Rumbaugh et al 1991) seems to be the most popular, and the most common symbols of this language are shown in Figure 10.

Some research has been conducted where the object-oriented model was used to design a temporal data schema, without embedding temporal constructs into the modelling language itself. One model is the multimodel and metamap schemas by Bjørnås and Skogan (1993) and Misund (1993), where spatial and non-spatial information can be accessed via parameters such as time.

Since the early 1990s we have seen increasing activity concerned with the design and construction of temporal object databases. Some ongoing research deals with incorporating concepts from these models into an object-oriented modelling language. Of particular interest is MADS (for Modelling of Application Data with Spatiotemporal features) which is described in Spaccapietra et al (1998) and Parent et al (1998). MADS is, so far, the most comprehensive modelling language for geographical and spatiotemporal information within the object-oriented paradigm. MADS provides the standard object-oriented concepts which can be marked with various icons to indicate certain spatial and temporal properties of these concepts.

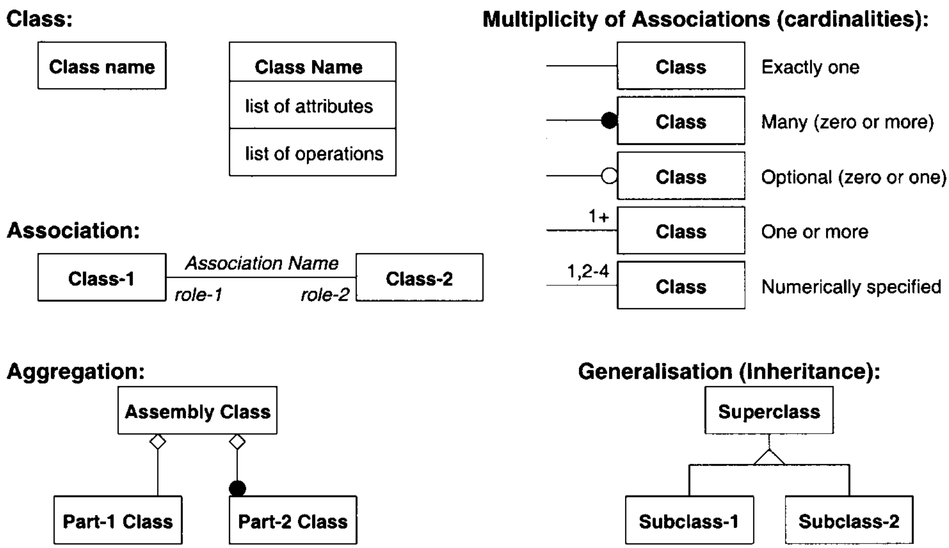

Figure 10 The most common symbols in the OMT-language.

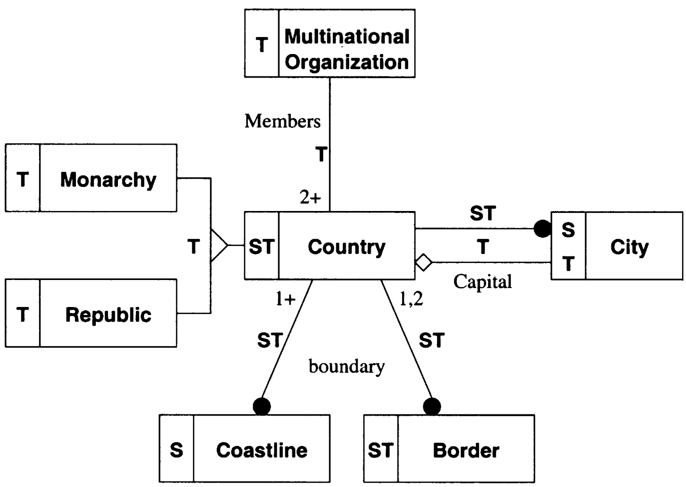

Figure 11 An Object Oriented model of a temporal map of Europe with temporal extensions.

A simple ERT-like extension of the OMT can also be found in Renolen (1999). Figure 11 shows an improved model of the countries of the world. Here, we have hidden the attribute information, but added a new type of object, viz. multinational organizations. A multinational organization has a number of member countries, and over time it may receive new members and existing members may secede from the organization. A country can also be a member of several such organizations. The markings on this diagram are defined as follows: a class that is T-marked has properties that vary over time. An object class that is S-marked is a spatial object and thus has a location. If an object class is ST marked, it means that the location of such objects may also vary over time. Hence, an object class may be both T-marked and S-

marked, but not **ST**-marked if it has a static location, but has other properties that vary over time. Subclasses inherit all markers from their superclasses, and are only marked if additional properties that deserve a mark are added to the object class.

A generalization link is **T**-marked if the instance of the subclasses may change in type, e.g. if a monarchy changes to become a republic. An aggregation or association link may be **T**-marked if the link only exists during a part of the time for which both involved objects co-exist. If, on the other hand, a link between two temporal object classes is not **T**-marked it means that the link exists as long as both objects co-exist. If an aggregation or association link is **S**-marked it means that the link is spatially dependent, i.e. is a topological link. Thus, a city can only be related to the country in which it is topologically inside.

Objects may have several properties that vary independently over time. Some properties may vary suddenly, such as the location and name of a country, whereas other properties may vary continuously, such as the population of countries and cities. Additional markers can be added to attributes if they are provided in the model.

7 Rules, Knowledge and Business Policy

In cartography, rules have mostly been applied to issues related to cartographic generalization. In conceptual modelling on the other hand, rules can be used to express constraints on conceptual schemas (such as cardinality constraints).

Most often, such rules are implemented directly in source code. In information systems, this represents potential problems since business policy may change over time, making current information systems obsolete (McBrien et al 1991). Historical data should always be viewed in context with the rules that were current at the time in question. For example, to become a member in the EU, a country has to present a national budget with a deficit that is less than a certain percentage of the gross domestic product. However, many countries that already are members of the EU, did not meet this requirement, but they still became valid members because the rule did not exist at the time when they became members.

A possible solution is to explicitly express rules by an *external rule language* (ERL) and store them together with the system. There are two approaches to this problem. One is to store the rules in a temporal database, such that rules can easily be obtained for specific points or periods of time. Another approach is to let the time validity be an inherent part of the rule if the rule is expressed as

$$\textbf{when}\langle \text{time}\rangle\ \textbf{if}\ \langle \text{condition}\rangle\ \textbf{then}\ \langle \text{consequence}\rangle$$

The rule can then be viewed as describing some logical constraint on the model, which must hold at every moment (or tick) whilst the information system is active (McBrien et al 1991).

Any system of rules must contain a set of predefined predicates that are understood by the system. In tense logic, four temporal predicates have been proposed (Rescher and Urquhart 1971). These are as follows:

$F(p)$: it will be that p
$P(p)$: it has been that p
$G(p)$: Henceforth, always p
$H(p)$: hereto-forth, always p

where p denotes any proposition. Similar constructs have been proposed by Alagic (1997) in a temporal constraint language as a high-level, declarative database programming paradigm for object-oriented databases.

In the field of artificial intelligence, rules have played a central role in the representation of knowledge about the real world. McCarthy's *situation calculus* (McCarthy and Hayes 1969), has been a major source of inspiration for other research in this area. In this theory, a *situation* is defined to hold the complete state of the universe at an instant of time, and the set S is defined to contain all situations including both real and imagined situations. A *fluent* is a function from S into S or into the set of booleans $B = \{true, false\}$. This means that a fluent $at(x, p, s)$ is true if an object x is at the position p in the situation s. Using the tense operators above, we can express causality. In order to express that in a situation s, if a person x is located at position p and it is raining at position p, then x will become wet, we could write:

$$\forall p \forall x \forall s \ \text{raining}(x, s) \wedge \text{at}(x, p, s) \rightarrow F(\text{wet}(p, s'), s)$$

Allen (1984) presents a further development of the situation calculus where events, processes, activities and causality are expressed as predicates occurring over time intervals along with binary operators over temporal intervals (Allen 1983). Consequently, if an object x moved from location *p1* to a location *p2* over a time interval t, this is expressed as the formula:

$$\text{OCCUR}(\text{CHANGE\_POS}(x, p1, p2), t)$$

8 Concluding Remarks

This article has presented a selection of conceptual modelling approaches and demonstrated some of their applications in the development of an STIS. This has made it possible to describe complex aspects of real world systems. A simple basic model has been presented, as shown in Figure 7, that recognizes three basic states of objects. On top of this underlying meta-model, there is a need to create application-specific domain models which precisely describe a domain in the real world. This model can be created by using ERT-models or object-oriented models together with data flow diagrams, behaviour models and rule-based constraints.

By using conceptual modelling languages, it is possible to describe and uncover the characteristics of real world systems in a way that is understandable even to non-experts. Thus, a broader audience can participate in the development of a system and the users will have a system that more closely represents their own concepts of reality.

Acknowledgements

This work was funded by the Research Council of Norway (NFR), grant 31387. Jan Terje Bjørke, Hans Hauska and Morten Dæhlen provided valuable comments on previous drafts. Thanks are also owed to the reviewers and to the Department of Surveying and Mapping at the Norwegian University of Science and Technology (NTNU).

References

Alagic S 1997 A temporal constraint system for object-oriented databases. In Gaede V, Brodsky A, Günther O, Srivastava O, Vianu D, and Wallace M (ed) *Constraint Databases and Their Applications*. Berlin, Spinger-Verlag Lecture Notes in Computer Science No 1191: 208–18

Allen J F 1983 Maintaining knowledge about temporal intervals. *Communications of the ACSM* 26: 832–43

Allen J F 1984 Towards a general theory of action and time. *Artificial Intelligence* 23: 123–54

Al-Taha K K and Barrera R 1990 Temporal data and GIS: An overview. In *Proceedings of GIS/LIS '90*, Anaheim, California, November 1990. Bethesda, MD, American Congress on Surveying and Mapping: 244–54

Al-Taha K K, Snodgrass R T, and Soo M D 1994 Bibliography on spatiotemporal databases. *International Journal of Geographic Information Systems* 8: 95–103

Austin J L 1962 *How To Do Things With Words*. Cambridge, MA, Harvard University Press

Birtwistle G 1979 *DEMOS: A System for Discrete Event Modelling on Simula*. London, MacMillan

Bjørke J T 1995 Digital kartografi, del 1. Unpublished Course Notes in Digital Cartography, Norwegian Institute of Technology

Bjørnås O M and David Skogan D 1993 Multimodels and spatio-temporal modeling in object-oriented GIS. Unpublished MS Thesis, Norwegian Institute of Technology

Booch G 1996 *Object Solutions: Managing the Object-Oriented Project*. Menlo Park, CA, Addison-Wesley

Chen P P 1976 The entity-relationship model: Towards a unified view of data. *ACM Transactions on Database Systems* 1: 9–38

Claramunt C and Thériault M 1995 Managing time in GIS: An event-oriented approach. In Clifford S and Tuzhilin A (ed) *Recent Advances in Temporal Databases*. Berlin, Springer-Verlag: 23–42

Coleman D, Hayes F, and Stephen Bear S 1992 Introducing objectcharts or how to use statecharts in object-oriented design. *IEEE Transactions on Software Engineering* 18: 9–18

Dahl O-J and Nygaard K 1966 SIMULA: An ALGOL-based simulation language. *Communications of the ACM* 9: 671–8

DeMarko T 1978 *Structured Analysis and System Specification*. New York, NY, Yourdon

Edwards J M and Henderson-Sellers B 1993 A graphical notation for object-oriented analysis and design. *Journal of Object-Oriented Programming* 5: 53–74

Gane C and Sarson T 1978 *Structured System Analysis: Tools and Techniques*. Englewood Cliffs, NJ, Prentice-Hall

Hadzilacos T and Tryfona N 1996 Logical data modelling for geographical applications. *International Journal of Geographical Information Systems* 10: 179–203

Hamre T 1995 Development of Semantic Spatio-temporal Data Models for Integration of Remote Sensing and In Situ Data in a Marine Information System. Unpublished PhD Dissertation, University of Bergen, Norway

Harel D 1987 Statechart: A visual formalism for complex systems. *Science of Computer Programming* 9: 231–74

Hornsby K and Egenhofer M J 1997 Qualitative representation of change. In Hirtle S C and Frank A U (ed) *Spatial Information Theory: A Theoretical Basis for GIS (COSIT'97)*. Berlin, Springer-Verlag Lecture Notes in Computer Science No 1329: 15–33

Hughes P H and Rønningen L A 1995 Demos activity digrams. Unpublished Course Notes in Discrete Simulation, Norwegian Institute of Technology

Käfer W, Ritter N, and Schöning H 1990 Support for temporal data by complex objects. In McLeod D, Sacks-Davis R, and Schek H-J (ed) *Proceedings of the 16th Conference on Very Large Databases (VLDB'90)*, Brisbane, Australia, 1990. New York, NY, Springer-Verlag: 24–35

Langran G 1992 *Time in Geographic Information Systems*. London, Taylor and Francis

Langran G and Chrisman N R 1988 A framework for temporal geographic information systems. *Cartographica* 25: 1–14

Martin J 1993 *Principles of Object-oriented Analysis and Design*. Englewood Cliffs, NJ, Prentice-Hall

McBrien P, Niézette M, Panatazis D, Seltveit A H, Sundin U, Theodoulidis B, Tziallas G, and Wohed R 1991 A rule language to capture and model business policy specifications. In *Advanced Information Systems Engineering, CAiSE'91*, Trondheim, Norway, May 1991: 307–18

McBrien P, Seltveit A H, and Wangler B 1992 An entity-relationship model extended to describe historical information. In *Proceedings of CISMOD '92*, Bangalore, India: 244–60

McCarthy J and Hayes P J 1969 Some philosophical problems from the standpoint of artificial intelligence. *Machine Intelligence* 4: 463–502

Medina-Mora R, Winograd T, Flores R, and Flores F 1992 The action workflow approach to workflow management technology. In Turner J and Kraut R (ed) *Proceedings of CSCW'92: Sharing Perspectives,* Toronto, Canada, November 1992: 281–8

Misund G 1993 Multimodels and metamaps: Towards an augmented map concept. Unpublished MA Thesis, University of Oslo

Montari A and Pernici B 1993 Temporal reasoning. In Tansel A U, Clifford J, Gadia S, Jajodia S, Segev A and Snodgrass R (ed) *Temporal Databases: Theory, Design, and Implementation*. Redwood City, CA, Benjamin-Cummings: 534–62

Parent C, Spaccapietra S, Zimanyi E, Donnini P, Plazanet C, and Vangenot C 1998 Modeling spatial data in the MADS conceptual model. In *Proceedings of the Eighth International Symposium on Spatial Data Handling,* Vancouver, Canada, July 1998: 138–50

Petri C A 1962 *Kommunikastion mit automaten*. Scrhiften des Rheinisch-Vestfallischen Institut für Instrumentelle Mathematik an der Universität Bonn

Peuquet D J 1994 It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. *Annals of the Association of American Geographers* 84: 441–61

Peuquet D J and Duan N 1995 An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographic data. *International Journal of Geographic Information Systems* 9: 7–24

Ramachandran B, MacLoad F, and Dowers S 1994 Modelling temporal changes in a GIS using an object-oriented approach. In Waugh T C and Healey R G (ed) *Advances in GIS Research: Proceedings of SDH'94*. London, Taylor and Francis: 518–37

Renolen A 1996 History graphs: Conceptual modelling of spatiotemporal data. Unpublished Paper Presented at GIS Frontiers in Business and Science, Brno, Czech Republic, 1996

Renolen A 1999 Concepts and Methods for Modelling Temporal and Spatiotemporal Information. Unpublished PhD Dissertation, Norwegian University of Science and Technology

Rescher N and Urquhart A 1971 *Temporal Logic*. New York, Springer-Verlag

Rumbaugh J, Blaha M, Premerlani W, Eddy F, and Lorensen W 1991 *Object-Oriented Modelling and Design*. Englewood Cliffs, NJ, Prentice-Hall

Searle J R 1969 *Speech Acts*. Cambridge, Cambridge University Press

Searle J R 1979 *Expression and Meaning*. Cambridge, Cambridge University Press

Segev A and Shoshani A 1993 A temporal data model based on time sequences. In Tansel A U, Clifford J, Gadia S, Jajodia S, Segev A and Snodgrass R (ed) *Temporal Databases: Theory, Design, and Implementation*. New York, NY, Benjamin-Cummings: 248–71

Sølvberg A and Krogstie J 1996 Information Systems Engineering: Advanced Conceptual Modeling. Unpublished Draft Manuscript

Sølvberg A and Kung D C 1993 *Information System Engineering: An Introduction*. New York, NY, Springer-Verlag

Spaccapietra S, Parent C, and Zimanyi E 1998 Modelling time from a conceptual perspective. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM),* Washington, D.C., 3-7 November 1998: 432–40

Teorey T J, Yang D, and Fry J P 1986 A logical design methodology for relational databases using the extended entity-relationship model. *ACM Computing Surveys* 18: 197–222

Theodoulidis C, Loucopoulos P, and Wangler B 1991 The entity relationship time model and the conceptual rule language. In Teorey T B (ed) *Proceedings of the Tenth Conference on the Entity Relationship Approach, San Mateo, CA, October 1991*. Pittsburgh, PA, ER Institute: 181–204

Wand Y and Weber R 1989 An ontological evaluation of system analysis and design methods. In

Falkenberg E D and Lindgren P (ed) *Information Systems Concepts: An In-Depth Analysis*. Amsterdam, North-Holland: 217–37

Worboys M F 1994a A unified model for spatial and temporal information. *The Computer Journal* 37: 26–34

Worboys M F 1994b Unifying the spatial and temporal components of geographical information. In Waugh T C and Healey R G (ed) *Advances in GIS Research: Proceedings of SDH'94*. London, Taylor and Francis: 505–17