An Application of Advanced Spatio-Temporal Formalisms to Behavioural Ecology

T. Ceccarelli[1], D. Centeno[2], F. Giannotti[1], A. Massolo[3], C. Parent[4], A. Raffaetà[5], C. Renso[1],

S. Spaccapietra[2] and F. Turini[1]

[1] KDDLab Pisa, ISTI CNR and Dipartimento di Informatica - Università di Pisa

[2] EPFL, Lausanne

[3] Dipartimento di Scienze Ambientali, Università di Siena

[4] Université de Lausanne

[5] Dipartimento di Informatica - Università Ca' Foscari di Venezia

Abstract. There is great potential for the development of many new applications using data on mobile objects and mobile regions. To promote these kinds of applications advanced data management techniques for the representation and analysis of mobility-related data are needed. Together with application experts (behavioural ecologists), we investigate how two novel data management approaches may help. We focus on a case study concerning the analysis of fauna behaviour, in particular crested porcupines, which represents a typical example of mobile object monitoring. The first technique we experiment with is a recently developed conceptual spatio-temporal data modelling approach, MADS. This is used to model the schema of the database suited to our case study. Relying on this first outcome a subset of the problem is represented in the logical language MuTACLP. This allows us to formalise and solve the queries which enable the behavioural ecologists to derive crested porcupines behaviour from the raw data on animal movements. Finally, we investigate the support from a commercial Geographical Information System (GIS) for the analysis of spatio-temporal data. We present a way to integrate MuTACLP and a GIS, combining the advantages of GIS technology and the expressive power of MuTACLP.

1 Introduction

Thanks to very low-cost modern sensing technologies and to the widespread use of mobile and ubiquitous communications based on GPS-equipped devices, geographic datasets about moving objects are growing rapidly. This opens new opportunities for monitoring and decision making applications in a variety of domains. Traffic control applications, for example, can replace traditional global traffic flow measurements with the precise follow-up of individual vehicles. The same applies to pedestrian flow in metro and railway stations or in commercial malls, allowing, e.g., for the evaluation of the best spots for services to passers-by. Human beings can be tracked on the basis of their cellular phone use, and fauna can be similarly tracked once equipped with micro-sensors. Natural phenomena can be tracked thanks to satellites. Such a wealth of space and time referenced data enables novel classes of applications with a potentially great social and economic impact. However, for these applications to become reality, new technical advances in information management are still needed. In particular, new user interfaces are requested in order to make data collection and management more easily available to application specialists. Another crucial factor is the availability of tools that enable consumable, concise and applicable knowledge to be extracted from the raw data. These tools will typically rely on analytical and reasoning processes that exploit the knowledge resource provided by spatial data warehouses. Unfortunately, current GIS (Geographic Information System) technology provides interfaces to computer specialists rather than to application experts. This has resulted in a significant slow-down in the development of new applications, poor data exchange capabilities, and major difficulties in re-using existing data. Moreover, spatial data warehousing is in its infancy, in particular when dealing with trajectories of moving objects and moving regions, such as pollution clouds, for example. This means that significant advances in spatio-temporal data management techniques are needed before society can take full advantage of the data that has become available.

This paper reports on a multi-disciplinary study jointly performed by behavioural ecologists and computer scientists. The aim is to experiment with the use of novel data management techniques in the monitoring of animal behaviour, which is a typical example of mobile object analysis.

Advanced database technologies have been used in order to represent, store and reason about the movements of animals. In particular, the paper describes how data about the movements of a number of crested porcupines, collected on the field, can be modelled, represented, and analysed via specific software systems in order to answer questions such as: “Where is the den of an animal?”; “Which animals form a couple?”; “What are the possible correlations between meteorological events (e.g., rain, snow, fire) and the movement of animals?”.

The problem has been addressed in two steps, each using a novel data management technique. First, we have designed a conceptual schema of the application data by using MADS [21], a data modelling approach which represents a state-of-art achievement for a user-oriented description of spatio-temporal data. Then MuTACLP [20], a spatio-temporal knowledge representation system, has been used in combination with a commercial GIS in order to compute the answers to the queries of interest. MADS (Modelling of Application Data with Spatio-temporal features) is an extended entity-relationship data model that provides many interesting features. These include the orthogonality of the structural, spatial and temporal modelling dimensions, the explicit description of spatial and temporal relationships, the explicit definition of aggregates, the generalisation/specialisation hierarchies, the support of both discrete and continuous views in space and time, and generic spatial and temporal abstract data types. MuTACLP (Multi-theory Temporal Annotated Constraint Logic Programming) is a formalism based on Constraint Logic Programming. It is designed to favour the construction of a software layer that supplies the user with a declarative spatio-temporal interaction with complex and mostly non declarative systems, such as heterogeneous databases, the web, and systems managing unstructured or semi-structured data. It allows for the representation and the handling of spatio-temporal information, and, at the same time, it allows knowledge to be organised in different modules which can be combined by means of meta-level composition operations.

Previous work on the development of conceptual and logical data models in the field of biology and ecology ranges from supporting taxonomic classifications [4] to the study of wildlife association and habitat analysis (see for instance [9]). The original contribution provided by this case study is twofold. On the one hand, behavioural ecology introduces spatio-temporal issues, which are of a more complex nature than those previously addressed in the literature. On the other hand the data modelling techniques provided by MADS and MuTACLP extend the currently implemented data modelling capabilities, as discussed below.

Regarding spatio-temporal data modelling, a historical overview and interesting insights for the future may be found in [24]. Out of the many spatio-temporal data models elaborated over the last decade, four of them are particularly well defined and aim at modelling data at a conceptual level, thus providing an approach that is comparable with MADS. ST-USM [15] and STER [33] build, like MADS, on the entity-relationship approach. Perceptory [2] and STUML [33] are based on the UML approach. Perceptory is a set of UML plug-ins that provide UML with the ability to handle space and time features. STUML is a spatio-temporal extension to UML. The functionalities provided by these models are similar to those of MADS, but their underlying principles are somewhat different. MADS, ST-USM, and STUML provide orthogonal concepts that can be freely combined, almost without restrictions, while STER and Perceptory restrict the possible combinations. For example, STER and Perceptory provide spatial objects (i.e., objects with a spatial extent), but no spatial attribute or relationship with a spatial extent. The designer using STER or Perceptory is forced to define as an object type any phenomenon with spatial features, even if from the application viewpoint the phenomenon would be naturally represented as an attribute or as a relationship type. This contradicts conceptual modelling principles. ST-USM and STER automatically enforce predefined constraints on the temporal extents of relationships and attributes to ensure that they are within the temporal extent of - respectively - the linked objects or the owner object. This is unfortunate as there are clearly cases in which such constraints contradict the application requirements. A peculiarity of ST-USM is that it supports the explicit definition of the granularities of the temporal and spatial extents. MADS supports valid time only, while ST-USM, STER and STUML provide both valid and transaction time. ST-USM shares with MADS a strong concern for providing a formal definition of the data model, thus eliminating any ambiguity. Finally, a peculiar feature of MADS is that it allows one to enrich relationships with a causal semantics (e.g., to express that objects of a given target type are generated from objects of a given source type) and with spatial and temporal constraints on the objects they link (e.g., topological constraints between the geometries of linked objects). It is also the only conceptual data model which has associated querying and manipulation languages.

As far as MuTACLP is concerned, some links exist with constraint databases [3, 7, 13]. In fact, from a database point of view, logic programs can represent deductive databases, i.e. relational databases enriched with intensional rules, constraint logic programs can represent constraint databases [17], and thus MuTACLP can represent spatio-temporal constraint databases. The spatio-temporal proposals in [3, 13] are extensions of languages originally developed to express spatial data only. As a consequence, the high-level querying mechanisms they offer are more oriented towards spatial data than towards temporal information. They can model only definite temporal information and there is no support for periodic, indefinite temporal data. On the contrary, MuTACLP provides several facilities to reason on temporal data and to establish spatio-temporal correlations. For instance, it allows one to describe *continuous* changes in time, as in [7], whereas both [3] and [13] can represent only *discrete* changes. Also indefinite spatial and temporal information can be expressed in MuTACLP, a feature supported only by the approach in [16].

The case study on crested porcupines has proved the effectiveness of MADS for helping behavioural ecologists in the design of databases which store all the information collected during their study. Similarly, it has proved that MuTACLP is an appropriate formalism to express queries related to the study of animals behaviour. Results from this case study generalise to the description and processing of any kind of mobile objects.

The paper is organised as follows. Section 2 gives some background on the case study. Section 3 presents the MADS data model and (part of) the conceptual schema that we developed for the case study. Section 4 describes the language MuTACLP, showing how some spatio-temporal queries relevant to behavioural ecologists can be implemented. Section 5 describes a customisation of a commercial GIS aimed at providing a useful support for our case study. Finally, Section 6 draws some conclusions and suggests future actions to extend current results.

2 The Crested Porcupine Case Study

The science of behavioural ecology studies animal behaviour in relation to the environment in which the animal lives. More specifically, our attention is geared towards the study of the behavioural ecology of the crested porcupine *Hystrix cristata*.

2.1 Background in Behavioural Ecology

The crested porcupine is a semi-fossorial rodent [28, 8, 18, 30]. The scarce information available on the behaviour of this species belonging to the gender *Hystrix* seems to indicate that the main activity of such animals during the time spent outside the den is feeding [8, 18, 30]. Clearly, most of the social interactions take place inside the den. This plays a critical role in terms of protection from predators, thermoregulation of the cubs, social behaviour and reproduction. The social unit is represented by extended family groups composed of 2-4 adults (or sub-adults) and by their cubs, which use the same dens [28, 11]. The choice of suitable sites for establishing a den seems to be conditioned by the availability of pre-existing cavities, by the pedological characteristics of the area and its climatic conditions [25]. Thus, dens may represent an important resource for the species and, as such, may influence the modalities of social distribution and aggregation.

The distribution and the abundance of resources, which are critical for a given species, influence its use of space and its modalities of social and spatial aggregation. The *home range* of an individual is thus a dynamic expression of its use of space. It can change over time as a consequence of variations in the age, in the reproductive and social state of an animal, or in the distribution and abundance of resources. According to recent studies on the behavioural ecology of crested porcupines [18, 30] conducted in the inner parts of Tuscany, the size of an annual home range can vary from 30 to 255 hectares.

The research in this context aims at understanding the social organisation and the variations in the activities and home ranges of crested porcupines. This is linked to resource (food, den sites, etc) distribution and abundance. More specifically, the goal is to evaluate the seasonal variations of home ranges of crested porcupines in a Mediterranean coastal area and to infer the factors which determine the size of the home ranges of these species. Other objectives of the research were: to investigate the existence of possible forms of spatial segregation or territoriality induced by differences in patterns of abundance and distribution of resources; to identify the features of the sites used as dens by the animals; and to verify the aggregation of animals in relation to the availability of dens and to variations in food resources.

2.2 Methods and Techniques

The study site is located in the Mediterranean coastal area of the Maremma Regional Park (Tuscany, Italy). Vegetation is characterised by dense scrub-wood, pine-wood, sparse pastures and cultivated lands (olive, maize, and sunflower). The data were collected between May 1998 and July 1999, as part of a long-term research on the behavioural ecology of crested porcupines. A number of animals were trapped by means of 14 double entrance box traps, positioned along the main trails used by crested porcupines. Traps were activated for at least 7 nights per month and checked at dawn.

In order to remotely localise the animals, a technique named radio-tracking, was employed. This has been commonly used in the field of Ethology and Animal Ecology since the Seventies. This technique is based on attaching radio-collars with individual frequencies to the animals. Subsequently the radio-tagged animals can be located at any time by identifying the source of the signal [14, 34]. The location of the animals can be calculated by triangulating the directions of maximum emission of the recorded signals by means of radio-receivers and antennas. These are located in a minimum of two stations. In our case the directional measurements were carried out from 78 stations selected within the study area, which have subsequently been geo-referenced.

The spatial location of an individual, enriched with the corresponding temporal information, is called a *fix*. The average number of locations for each month and for each individual varied from 20 to 50. In a global time period of two years about 5000 fixes were collected. It is worth noting that telemetry measurements are subject to the influence of many error sources: errors of the measurement devices and operators, and errors related to the environmental conditions. In our case, the location error was assessed by positioning a number of radio-collars in known positions and by calculating the gap (in meters) between the real and estimated position and the difference between the estimated and real bearings. The estimations resulted in being -/+ 5° for the bearings, and 62 meters for the position. Despite the fact that new GPS technologies are nowadays rapidly progressing, telemetry still has an important role in Ethology and Animal Ecology. The small dimensions of radio-transmitters, their limited weight and their low-cost make it unlikely that they will be replaced, at least for the next decade. Given the widespread use of this technique in the study of several animal species, the need to obtain increasingly precise measurements and consistent error estimates has emerged.

For each animal monitored the annual and seasonal home range has been calculated applying the method of Minimum Convex Polygon (MCP) [34]. This is one of the oldest and simplest methods to compute the home range, and it is based on the calculation of the minimum convex polygon that includes all the localisations of the animal. The weakest aspect of this method is the presence of *outliers*, that is, localisations that are far from all the others. These influence the extent of the polygon, which therefore includes large areas never used by the animal. To alleviate this problem the Fixed Kernel method was applied for the calculation of the usage distribution [35]. This is one of the most powerful probabilistic techniques in order to compute the home range area with different probabilities.

Finally, spatial analyses was performed using ArcView 3.2 by ESRI [10] and a number of extensions (Spatial Analyst, 3D Analyst, Patch Analyst and Movement version 2.0, by USGS Alaska). This choice was made on the basis of several factors, such as the in-house availability of the software, a user-friendly interface, and the existence of the extension *Movement* which allows home range calculations and other spatial analyses in animal ecology.

2.3 Spatio-Temporal Questions From The Application Domain

Given the above mentioned general goals, here we focus on a number of problems and queries specifically related to spatio-temporal aspects which cannot be solved in a simple way by using a standard GIS. Each problem is expressed from two different stances: the biological stance in finding a solution to the question, and, computationally, the expected behaviour of the software tool (GIS and/or DBMS) in resolving the query. In details:

Problem (1) Den Localisation. Den localisation is carried out by behavioural ecologists on a bi-monthly basis using a procedure called *homing in*. The ecologist follows the radio-signal of individuals during the day to *physically* locate the dens of the animals. Each time a den is found, it is recorded on a database. This method is very expensive and cannot always be applicable. Furthermore, animals usually change their dens, and as such den localisation of some animals is not known in those periods not covered by homing in. Thus a strong concern consists in finding the den location when no information is otherwise available.

Query. Given a number of known den positions (collected only on a bi-monthly basis) and the animal fixes, infer the position of dens in periods for which no information is available.

Functionalities Expected From GIS/DBMS. The problem of den localisation requires a system that, starting from the time interval of interest for the analysis, is capable of automatically computing locations that are likely to be a den. The user should have the option of selecting the individual, the period of interest and the probability threshold for a den.

According to domain experts' opinion, crested porcupines stay inside or close to their den from dawn to sunset. Hence, the function determining a den should select only fixes collected in this specific time interval and analyse such data grouping them by taking into account both the spatial and temporal dimensions. As a result, the system returns a set of possible dens, i.e. coordinates of locations that can represent a den with a certain probability.

Problem (2) Relations among Animals. In order to understand the habits and social behaviour of animals it is extremely important to discover the relationships existing between individuals.

To assess the degree of association between individuals, one way would be to compare the overlap of their home ranges, estimated at given time intervals (typically every one month or, more generally, k months). However, such a method is quite raw because it could be that animals stay in common areas but at different periods of time. Moreover the overlapping area might be so large that the simultaneous presence of the animals in such an area does not ensure that they are really close to each other.

A more precise procedure requires the calculation of the inter-individual distance between animals localised at the same time. We say that two fixes are *contemporary* if they refer to localisations of animals in the same place and at the same time, i.e., we consider a kind of spatio-temporal closeness between individuals. Since the tracking technique usually presents several sources of error, in the analysis two fixes are assumed to be *contemporary* if they fall within a given time interval and the corresponding positions are within a certain distance. The effective values for the temporal and spatial thresholds are established (and can be varied) by behavioural ecologists.

By analysing this kind of inter-individual distance between animals it is possible to make hypotheses about which animals can be considered a couple, which ones form a herd, or which individuals avoid some others.

Query. Given the animal fixes, determine whether two animals are likely to be a couple by using the above inter-individual distance.

Functionalities Expected From GIS/DBMS. This problem seems to be very "heavy" from a computational point of view, when standardised and automated procedures are not applied. The use of DBMS/GIS should help in solving the problem, provided that accurate procedures are developed in order to identify contemporary intervals, estimate the required metrics (distance between points), and to produce data to be used in subsequent statistical analyses. The condition for identifying contemporary fixes should be definable by the user. Furthermore, the user should have the option of performing calculations on a subset of the original data by selecting a time period and/or animal identifiers.

Problem (3) Spatio-Temporal Distance Between Localisations And Events. The spatio-temporal distance between an animal and a particular event, defined in time and space, is one of the problems which is often addressed by animal ecology researchers. Indeed often, it is necessary to assess whether and to what extent an event (e.g. a change in a crop cover, a hunting chase, a meteorological or geo-morphological occurrence) defined in time and space, has caused variations in the habits of the monitored animals. Notice that typically spatial and temporal limitations of the effect of the event on animals must be specified. For instance, we could assume that events taking place at a spatial distance from the localisation of the individual greater than a fixed threshold cannot influence it. Similar considerations apply to the temporal distance for events occurring before the localisation. In addition events occurring later the localisation should not be taken into account. We should expect that if an event concretely influences the animals, it would then "attract" or "reject" them, to an extent which is proportional to their spatio-temporal distance. This technique could also be useful in order to estimate the time interval between the occurrence of the event and its effects (as well as the duration of such effects) on the animal.

Query. Given a dataset with spatio-temporal locations of all monitored animals during the period under study and a dataset referring to a series of events defined in space and time, determine whether the events influence the animals.

Functionalities Expected From GIS/DBMS. DBMS and GIS are expected to allow the user to discover complex spatial and temporal relationships between the event set and the animal localisation set. At first, some parameters have to be selected, e.g. which animals and events have to be studied. Then, it is particularly important to define the spatio-temporal constraints determining the interval of influence of events upon the animals. In fact, as mentioned above, depending on the specific case, localisations too distant (in time and/or in space) from the occurring event may not be affected by the event itself and should be automatically excluded from the analysis. To this end a number of parameters are to be set, e.g.:

- the maximum *time* distance between the animal localisation and the selected event,
- the *spatial* distance threshold between the event and the animal localisation.

2.4 Data Organisation and Conceptual Modelling

In wildlife research, a large amount of field data is collected: many environmental variables are monitored in order to infer the influences of their change on behavioural patterns. Furthermore many factors that could influence animal choices may not have been fully taken into consideration from the beginning of the study; as a consequence, a large set of data is usually collected in order to take into account possible variations in the research design protocol.

For this reason the extent and structure of the data collected is very often not clear from the onset. This becomes even more evident when trying to formalise further research issues in terms of a conceptual data model. Nevertheless, a conceptual data description has been developed a posteriori (i.e. after most of the data collection had already been undertaken). This kind of reverse engineering of existing data is frequently needed and is known to be extremely useful in order to have a better understanding of the application data and of the results that can be achieved through data processing.

3 MADS

MADS [21, 22] is a spatio-temporal conceptual data model. It covers four facets (or dimensions) of the data modelling process: data structure, space, time, and perception. Data structures are modelled according to the well-known extended entity-relationship paradigm (EER, whose major commercial representative is UML). That is to say, a database schema is a graph of object types connected by relationship types. Object types and relationship types are described by properties (attributes and methods) that define their static and behavioural characteristics of interest for the application. MADS has no limitation on attributes which can bear single or multiple values as well as atomic or composite values. Similarly, MADS has no limitation on relationships, which can be of any arity (e.g., binary, ternary) and can bear properties. Object types as well as relationship types are organised into generalisation/specialisation hierarchies to provide support for classification refinement. Cardinality constraints are part of the data model. For attributes they are used to rule the number of possible values an attribute may hold while for relationship roles they rule the number of relationships an object in a given role can participate in. MADS is currently supported by a prototype schema editing CASE tool, which provides the user with an intuitive visual interface for the definition of MADS schemas. Once the definition of a schema is completed, the tool automatically translates it into a logical schema suitable for its implementation onto the existing DBMS or GIS chosen by the user [23].

In the following we present the most significant features of MADS. However, concepts for supporting multiple perceptions are not discussed in this paper. Finally, we present and explain a subset of the MADS schema we have developed for the crested porcupine monitoring application.

3.1 Structural Dimension

Long discussions with designers of geographical applications have led us to extend MADS structural capabilities beyond those typical of the EER paradigm. A first extension allows relationships to link groups of objects, instead of individual objects. For example, in a cartographic application two groups of building representations can be related because they portray the same set of real world buildings at different scales. Note that in this situation relating individual building representations would not make sense. MADS uses the concept of *multi-association* to express relationships among sets of objects. Thus, a relationship type is either a (normal) association (between individual objects) or a multi-association.

Keeping track of object evolution is another requirement that emerged from applications. To cope with this, MADS offers the possibility to attach specific semantics to relationships. A *transition* semantics is attached to a relationship type whenever it models the fact that objects from a *source* subclass move into a *target* subclass. For example, a building acquired by a public administration moves from the *PrivateBuilding* class to the *PublicBuilding* class. Instances linked by a transition relationship bear the same object identifier. Similarly, a *generation* semantics is attached to a relationship type whenever it models the fact that objects from a *source* class produce objects in a *target* class. For example, the reorganisation of a set of land plots produces a new set of land plots. In this example, the same object type, *LandPlot*, serves as source and target of the generation, and the relationship type is of a multi-association kind. Objects linked by a generation relationship have different object identifiers.

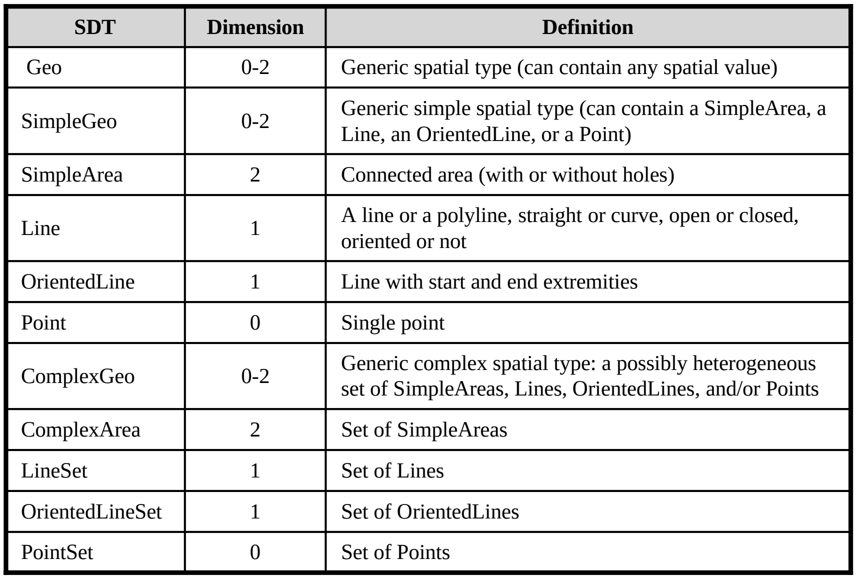
Relationship types can also be given an *aggregation* semantics, to denote that the link relates a composite object to one of its component objects. For example, an *Individual* crested porcupine may be related to a crested porcupine *Family* through the aggregation relationship *Belong* in order to express that this individual is a member of this family.

Finally, beyond extending generalisation/specialisation hierarchies to relationship types, MADS precisely defines all forms of inheritance that may be used to control propagation of properties from the supertypes to the subtypes. In particular, MADS defines the exception mechanisms known as *inheritance refinement*, *redefinition*, and *overloading*, which turn out to be important when dealing with spatial and temporal data.

3.2 Spatial Dimension

MADS concepts allow us to model spatial information according to both the discrete and continuous view of space.

From the discrete view, traditional data is complemented, whenever appropriate, with a spatial extent. To avoid unnecessary constraints on the design of a database schema, MADS allows spatial extents to be associated with objects as well as with attributes and relationships. Values for spatial extents are driven by a set of predefined spatial data types, added to the usual set of data types for alphanumeric databases. The MADS spatial data types, illustrated in Fig. 1, are similar to those promoted by the Open Geodata Consortium. They include the obvious *Point*, *Line*, and *Area* data types and their set equivalent (*PointSet*, *LineSet*). They also include generic types (*Geo*, *SimpleGeo*, and *ComplexGeo*). These data types allow, for example, objects of the same object type (or values of the same attribute) to acquire spatial values that belong to different subtypes in the data type hierarchy. For example, an object type *Lake* with *SimpleArea* extent and an object type *River* with *Line* extent can share a common supertype, *WaterExtent.* In this case



SDT	Dimension	Definition
Geo	0-2	Generic spatial type (can contain any spatial value)
SimpleGeo	0-2	Generic simple spatial type (can contain a SimpleArea, a Line, an OrientedLine, or a Point)
SimpleArea	2	Connected area (with or without holes)
Line	1	A line or a polyline, straight or curve, open or closed, oriented or not
OrientedLine	1	Line with start and end extremities
Point	0	Single point
ComplexGeo	0-2	Generic complex spatial type: a possibly heterogeneous set of SimpleAreas, Lines, OrientedLines, and/or Points
ComplexArea	2	Set of SimpleAreas
LineSet	1	Set of Lines
OrientedLineSet	1	Set of OrientedLines
PointSet	0	Set of Points

Fig. 1. MADS Spatial data types

WaterExtent will bear the generic *SimpleGeo* data type, which is a super-type of both *SimpleArea* and *Line*. The generalisation/specialisation hierarchy of the spatial data types is shown in Fig. 2.

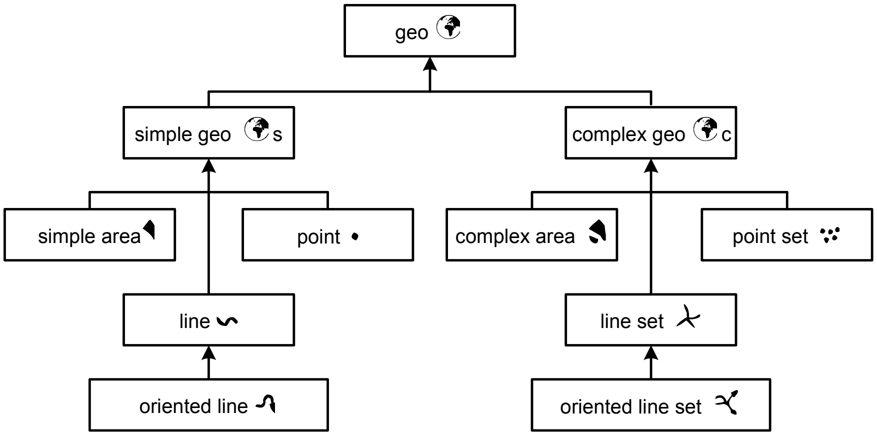
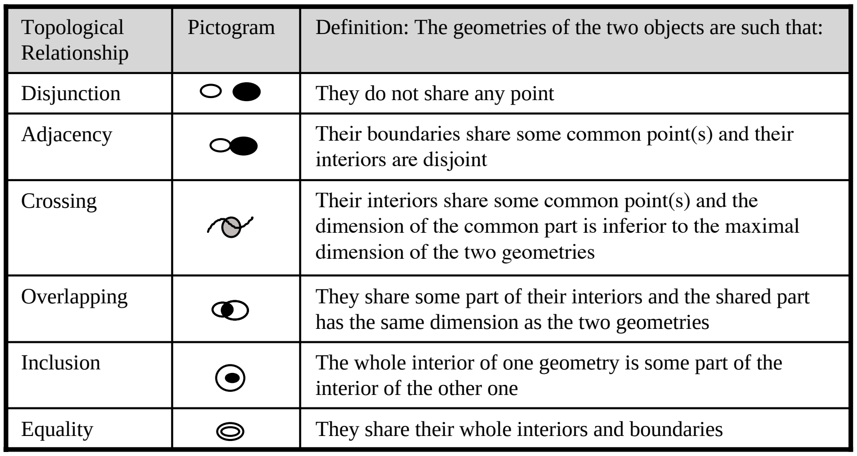


Fig. 2. MADS spatial data type hierarchy

For a more precise representation of spatial phenomena, MADS supports a predefined set of topological relationships (see Fig. 3). They can be used within the definition of a schema to attach a topological semantics to a relationship type. For example in Fig. 6, the topological relationship



Topological Relationship	Pictogram	Definition: The geometries of the two objects are such that:
Disjunction	○ ●	They do not share any point
Adjacency	○●	Their boundaries share some common point(s) and their interiors are disjoint
Crossing		Their interiors share some common point(s) and the dimension of the common part is inferior to the maximal dimension of the two geometries
Overlapping	◐○	They share some part of their interiors and the shared part has the same dimension as the two geometries
Inclusion	⦿	The whole interior of one geometry is some part of the interior of the other one
Equality	◎	They share their whole interiors and boundaries

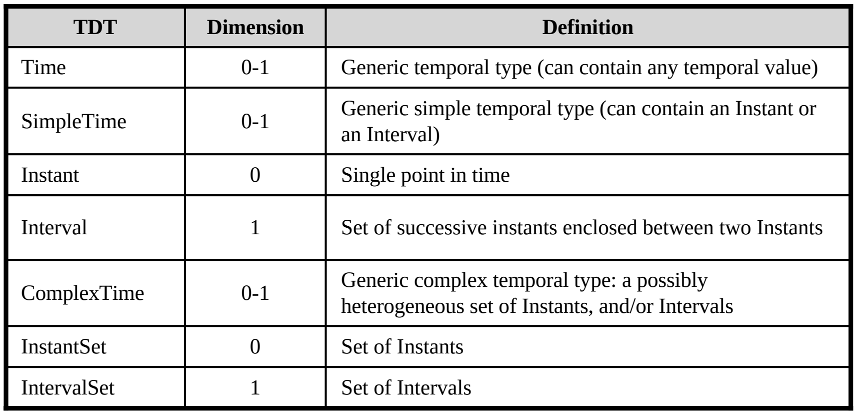
Fig. 3. Topological relationships

IsFound, between the object types *Territory* and *Burrow*, bears an inclusion constraint enforcing the rule that a given burrow of a crested porcupine may be linked to a given territory only if the extent of the burrow (a point) is inside the extent of the territory (a simple area).

Differently from the discrete view, the continuous view of space needs to be able to associate information with a spatial extent (a geographical zone), rather than with an identifiable object. For example, some applications may need to store altitude or temperature in a given region, which may either be the whole extent covered by the database (denoted as DBSpace) or an extent associated with an object in the discrete view. For example, in a national geographical database altitude and temperature may be stored all over the country, or only for major cities. MADS allows us to describe continuous fields as space-varying attributes. A space-varying attribute is an attribute whose value is defined by a function whose domain is some part of the space (DBSpace, or any other identifiable extent) and whose range is a given domain of values (for instance, integers for altitude). Space-varying attributes are denoted in MADS schema diagrams using the icon `f(🌍)`.

3.3 Temporal Dimension

As most geographical applications need to record data over some period of time, MADS also supports temporal modelling. To reduce the complexity for users, MADS exploits as much as



TDT	Dimension	Definition
Time	0-1	Generic temporal type (can contain any temporal value)
SimpleTime	0-1	Generic simple temporal type (can contain an Instant or an Interval)
Instant	0	Single point in time
Interval	1	Set of successive instants enclosed between two Instants
ComplexTime	0-1	Generic complex temporal type: a possibly heterogeneous set of Instants, and/or Intervals
InstantSet	0	Set of Instants
IntervalSet	1	Set of Intervals

Fig. 4. MADS Temporal data types

possible the many similarities between space and time. For example, a hierarchy of temporal data types, illustrated in Fig. 4 and Fig. 5, parallels the hierarchy of spatial data types and provides the basic data types (*Instant* and *Interval*), their set counterparts, and the generic *Time* type.

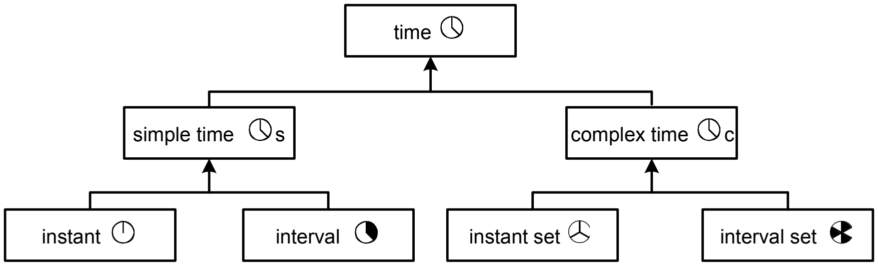


Fig. 5. MADS Temporal data type hierarchy

The temporal extent, possibly associated with object and relationship types, conveys the information on the lifecycle of an object or a relationship. The lifecycle of an instance tells when the instance is created, activated, suspended, reactivated, and deleted. For example, defining *Individual* in Fig. 6 as a temporal object type allows us to record the lifecycle of each crested porcupine, i.e. the time interval from its estimated date of birth up to its estimated date of death or the current time if it is not dead. A lifecycle may also be made up of active and suspended periods. For

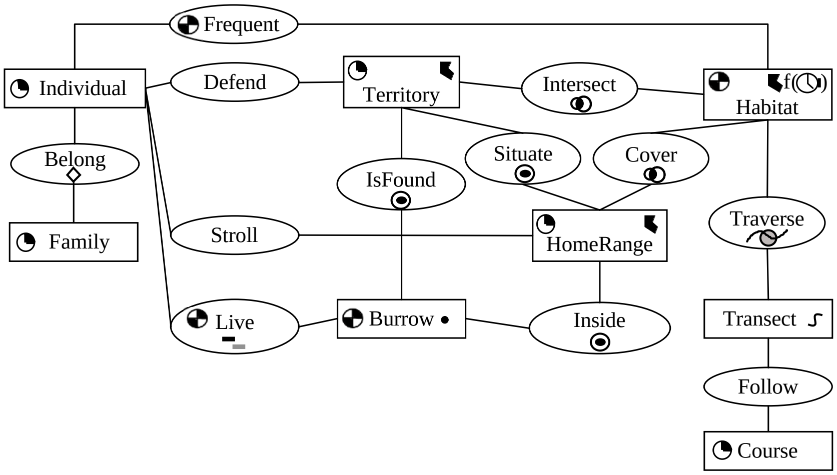


Fig. 6. Simplified diagram of the MADS crested porcupine database

example, the lifecycle of the *Burrow* object type describes when the burrow is inhabited (active) and when it is empty (suspended).

Lifecycles in MADS are meant to convey valid time, i.e. to describe when the objects and relationships are active from the application viewpoint.

Evolution in time of the value of an attribute parallels a continuous field in space and can be recorded by defining the attribute as being time-varying, a fact which is denoted in MADS diagrams by the icon f(◷). A time-varying attribute is an attribute whose value is defined by a function whose domain is some time extent and ranging over a given class of values. For example, as crested porcupines may add new entries to their burrows, the *entry* attribute of *Burrow* in Fig. 7 is time-varying: the database will keep track of the evolution of the set of entries of each burrow. In the same way, in Fig. 6, the evolution of the geometry (an area) of the *Habitat* object type is maintained.

Similar to topological relationships in space, MADS supports *synchronisation* relationships in time (e.g., after, before, during). They define constraints on the lifecycles of the linked objects. For example, the *Live* relationship type has an overlap synchronisation semantics to ensure that an individual crested porcupine may be linked to a burrow only if the lifecycles of the individual and burrow overlap.

3.4 A Schema for the Crested Porcupine Database

Modelling the crested porcupine case study with MADS has resulted in the schema illustrated in Fig. 6 (for readability, the figure only shows types, with no details of attributes and methods). For instance, the possibility of modelling complex objects, with multiple levels of decomposition in the attribute structures, has allowed us to represent trapped crested porcupines as a single object type (with the relational data model this would have required the use of several tables). Temporal concepts are used to model the lifecycle of several objects (e.g., a crested porcupine is modelled as a temporal object type *Individual*), while spatial concepts represent the location of objects (e.g., points for burrows). Space and time are combined to model time-varying geometry for objects such as *Habitat*. Topological relationships enforce spatial constraints. For instance a burrow can be linked to a territory through the topological relationship *IsFound* only if the location of the burrow is inside the extent of the territory. Similarly, the synchronisation relationship type *Live*, allows us to link a crested porcupine (Individual object type) to a burrow only if the two lifecycles overlap.

The crested porcupine database schema is quite complex, as is the real world it represents. For readability, in the presentation we decompose it into two sub-schemas, gathering the information needed for a specific sub-domain of interest: what relates to the social activities of the animals, and what relates to their home ranges.

The social behaviour is modelled in Fig. 7 by using the following object and relationship types:

- *Individual* represents an adult, radio-collared, crested porcupine for which data have been collected. It is a temporal object type whose lifecycle is a time interval ranging over the (estimated) existence of the animal. It is worth noting that usually the animal lifespan is not precisely known. An extension of the data model to support imprecise time specification would be useful here.
- *Family* represents a couple of porcupines with their cubs. A set of animals is classified as a family by exploiting the fact that a couple and their cubs are often found together in the same

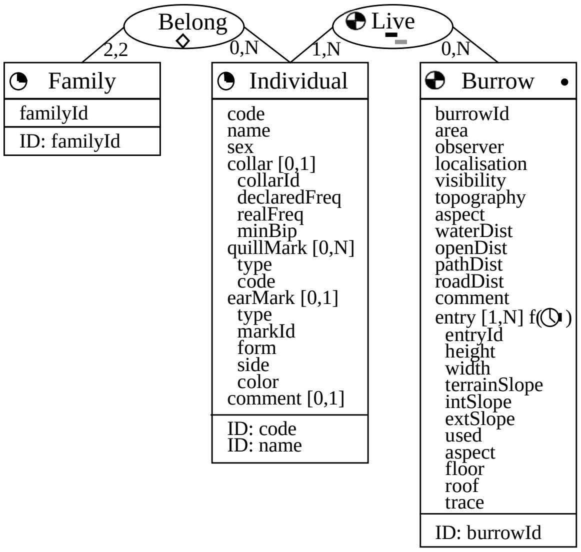


Fig. 7. The social sub-schema

burrow and wandering together. They may also be trapped together. Family is a temporal object type whose lifecycle is a time interval stating when the couple exists.

- *Burrow* represents porcupines dens. It is a spatial object type with geometry of type Point. Its lifecycle represents the set of time intervals during which the burrow is inhabited.
- *Belong* is an aggregation (shown in the diagrams by a diamond) describing the two adults of a family.
- *Live* is a temporal relationship stating which crested porcupines live in which burrows and when. "When" is described by the set of time intervals of the lifecycle of the Live relationship. Live is also a synchronisation relationship of kind *overlapping*: it prevents users from linking individuals and burrows that do not exist simultaneously.

Features concerning home range are shown in Fig. 8. The home range subschema shares with the previous social behaviour subschema the *Individual* and *Burrow* object types. The semantics of the other object and relationship types of the home range subschema are as described below:

- *Habitat* represents an area with a specific kind of plant, e.g. a pine tree forest, a pasture, or a cultivated land plot. Its lifecycle is a set of time intervals which describe when these plants

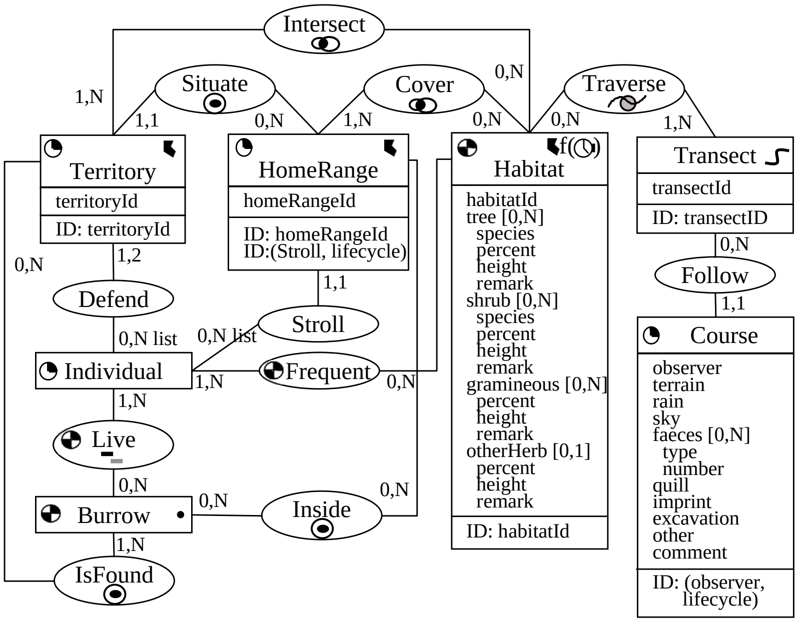


Fig. 8. The home range sub-schema

can be found in this area (e.g., a corn land plot is bare during Winter). The geometry of a Habitat is an area that can evolve through the years (e.g. a part of a forest may be chopped down): its geometry is time-varying. It is represented in Fig. 8 by the symbol f(◷).

- *Transect* represents a route that is usually followed by the observers when they collect data about crested porcupines. It is a spatial object type whose geometry is a line.

- *Course* represents the fact that some nights an observer follows a Transect in order to observe crested porcupines. The Course object type stores the findings collected by the observer. Course is a temporal object type, whose lifecycle is a time interval describing when it takes place.

- *HomeRange* is defined for a crested porcupine and a period of time (usually a season or year). It represents the area that the animal uses in the given period, i.e. the area where it has been observed or captured during this period. It is a spatial and temporal object type: its lifecycle is the time interval for which the HomeRange is computed. Its geometry is the area used by the crested porcupine.

- *Territory* is defined for a crested porcupine (or a family) and for a period of time. It is the area that in the given period of time the animal (or the couple) defends as its own private area, by preventing other crested porcupines from going there. Roughly, Territory is computed by subtracting from the HomeRange of the animal (or family) all the homeranges of the other crested porcupines. It is a spatial and temporal object type: its lifecycle is the time interval for which the Territory is computed while its geometry is the area defended by the crested porcupine.

  The last two object types, as well as the relationship types linking them, are computed from the measured positions of the crested porcupines.

- *Traverse* is a topological relationship type of kind *Crossing* that links a transect to the habitats that it traverses.
- *Follow* is a relationship type linking a transect to the course followed by the observer during that night.
- *Frequent* is a temporal relationship type linking a crested porcupine and a habitat. It states when the animal has been observed in this habitat. It is computed from the stored locations and the geometry of the habitats. The lifecycle of Frequent is a set of time intervals.
- *Stroll* is the relationship type that links a crested porcupine to its successive home ranges.
- *Defend* is the relationship type that links a crested porcupine to its successive territories.
- *Situate* is a topological relationship type of kind Inclusion enforcing that, at each given instant, the territory defended by a crested porcupine is inside its home range.

3.5 Summary Assessment

The schema that we developed for the case study application allows us to formulate some comments on the benefits achieved by using MADS versus using another modelling approach.

The most obvious and already well-known advantage is in terms of the compactness and readability of the resulting conceptual schema. This is due to MADS EER (Extended Entity Relationship) background. MADS adopts EER facilities, such as n-ary relationships with attributes, complex objects (i.e., objects with attributes composed of attributes), and multivalued attributes. For example, Fig. 7 shows the object type *Individual* holding three composite attributes, *collar*, *quillmark* and *earmark*, which gather a total of 11 atomic attributes. The attribute *quillMark*, as the attribute *entry* in *Burrow*, is a multivalued attribute. While the application schema shows only binary relationship types, several show role cardinalities of type n-m. All these features are not supported in the relational data models that equip current DBMS and GIS software. This establishes an undisputed superiority of EER modelling versus relational modelling in terms of conceptual schema design.

In comparison with other spatio-temporal conceptual data models, MADS offers the possibility to attach both space and time features to every item in a schema, be it an object, an attribute, or a relationship. The figures easily show that this possibility has been extensively used within the case study, in particular to define spatial and temporal object and relationship types. The attribute *entry* in *Burrow* is an example of a composite time-varying attribute. While there is no example of spatial attribute, it is easy to see that in Fig. 8 *Course* could have been a spatial attribute of the object type *Transect*, should the designer have decided to model course as an attribute instead of as an object type. Finally, MADS is the only spatio-temporal conceptual data model that allows to attach topological and synchronisation semantics to relationships between object types, as shown in Fig. 7 and Fig. 8.

4 MuTACLP: a logical language to support reasoning on animal behaviour

In the previous section we showed how a conceptual model for the crested porcupine case study can be formalised in MADS. Such a model provides an explicit representation of the relevant entities toghether with their properties, and of the relationships between them. In this section we introduce the language MuTACLP [20, 1] and its reasoning capabilities. Using this and also by relying on the understanding provided by the conceptual model, we can answer some of the queries of interest for behavioural ecology.

MuTACLP has been proposed as a language for representing and handling spatio-temporal information in a framework where pieces of knowledge can be encoded in different programs and possibly combined together by means of composition operators. In [27, 19] is shown, by means of several examples, how MuTACLP can be used to improve the spatio-temporal analysis of geographical data. On the one hand, both temporal and spatial information can be represented directly in MuTACLP. On the other hand, MuTACLP can also be used to express (temporal) knowledge on spatial data stored in a commercial GIS. In this second case, the language allows the user to directly access the specific functionalities provided by the GIS which are thus "combined" with the high-level reasoning capabilities of MuTACLP.

In Section 4.1 we describe the language MuTACLP while in Section 4.2 we demonstrate how MuTACLP can be useful to support spatio-temporal analysis for the crested porcupine case study. As pointed out in Section 2.3, many questions concerning the mating system and the social structure of this animal are still unanswered. Such questions often involve both temporal and spatial aspects, which are usually strictly correlated. For this reason they are difficult to express using current GIS technology since it does not provide high-level operations to support spatio-temporal queries. On the contrary, by using our language we obtained some encouraging results. These suggest how suitable this language is both for the formulation of some of the queries of interest and for the reasoning activity needed to obtain lacking information.

4.1 Multi-theory Temporal Annotated Constraint Logic Programming

Next, we present MuTACLP (Multi-theory Temporal Annotated Constraint Logic Programming) [20, 19], a language which allows us to represent temporal information by means of annotations and spatial data by using constraints in the style of the constraint databases approaches [3, 7, 13].

Time Domain and Annotated Atoms. Let us start by describing the temporal domain underlying MuTACLP. Time can be discrete or dense. Time points are totally ordered by the relation $\leq$. The set of time points, denoted by $\mathcal{T}$, is equipped with the usual operations (such as $+$, $-$). We assume that the time-line is left-bounded by 0 and open to the future, with the symbol ∞

used to denote a time point that is later than any other. A *time period* is an interval $[r, s]$, with $0 \leq r \leq s \leq \infty$, $r \in \mathcal{T}$, $s \in \mathcal{T}$, that represents the convex, non-empty set of time points $\{t \in \mathcal{T} \mid r \leq t \leq s\}$[1]. Thus the interval $[0, \infty]$ denotes the whole time line.

Annotated formulae, the basic constituents of MuTACLP programs, are of the form $A\,\alpha$ where A is an atomic formula and α is an annotation. We consider three kinds of annotations based on time points and time periods. Let t be a time point and let $J = [r, s]$ be a time period. Then

(`at`) The annotated formula $A\,\mathtt{at}\,t$ means that A holds at time point t.

(`th`) The annotated formula $A\,\mathtt{th}\,J$ means that A holds *throughout*, i.e., at *every* time point in the time period J. The definition of a `th`-annotated formula in terms of `at` is:

$$A\,\mathtt{th}\,J \;\Leftrightarrow\; \forall t\ (t \in J \rightarrow A\,\mathtt{at}\,t).$$

(`in`) The annotated formula $A\,\mathtt{in}\,J$ means that A holds at *some* time point(s) - but we may not know exactly which - in the time period J. The definition of an `in`-annotated formula in terms of `at` is:

$$A\,\mathtt{in}\,J \;\Leftrightarrow\; \exists t\ (t \in J \wedge A\,\mathtt{at}\,t).$$

The `in` temporal annotation accounts for indefinite temporal information.

Partial Order and Constraint Theory. The set of annotations is endowed with a partial order relation $\sqsubseteq$. Given two annotations α and β, the intuition is that $\alpha \sqsubseteq \beta$ if α is "less informative" than β in the sense that for each formula A, $A\,\beta \Rightarrow A\,\alpha$. More precisely, in addition to Modus Ponens, we consider two further inference rules, i.e., the rule $(\sqsubseteq)$ and the rule $(\sqcup)$ below.

$$\frac{A\,\alpha \qquad \gamma \sqsubseteq \alpha}{A\,\gamma}\ \textit{rule}\ (\sqsubseteq) \qquad\qquad \frac{A\,\alpha \qquad A\,\beta \qquad \gamma = \alpha \sqcup \beta}{A\,\gamma}\ \textit{rule}\ (\sqcup)$$

The rule $(\sqsubseteq)$ states that if a formula holds with some annotation, then it also holds with all smaller annotations. The rule $(\sqcup)$ says that if a formula holds with two annotations α and β, then it holds with the least upper bound $\alpha \sqcup \beta$ of such annotations. For technical reasons related to the

[1] The results we present naturally extend to time lines that are bounded or unbounded in other ways and to time periods that are open on one or both sides.

properties of th and in annotations the application of the rule ($\sqcup$) is restricted to th annotations with overlapping time periods and it returns a th annotation with an interval which is the union of the two overlapping time periods (see [1] for details).

The axiomatisation of the partial order relation on temporal annotations is contained in a constraint theory. Briefly, according to the intuitive meaning of the relation $\sqsubseteq$, we have $\mathtt{th}\,[r_1, r_2] \sqsubseteq \mathtt{th}\,[s_1, s_2]$ if and only if the time period $[r_1, r_2]$ is a subinterval of $[s_1, s_2]$, whereas $\mathtt{in}\,[r_1, r_2] \sqsubseteq \mathtt{in}\,[s_1, s_2]$ if and only if $[r_1, r_2]$ is a superinterval of $[s_1, s_2]$. The constraint theory also includes the axiomatisation of the greatest lower bound $\sqcap$ of two annotations which is needed when composing programs. For a detailed description of the constraint theory we refer the reader to [1].

MuTACLP Programs and Composition Operators. A MuTACLP program is a finite set of MuTACLP *clauses*, i.e., of clauses of the form:

$$A\,\alpha :- C_1, \ldots, C_k, B_1\,\alpha_1, \ldots, B_n\,\alpha_n$$

where $A, B_1, \ldots, B_n$ are atoms, $\alpha, \alpha_1, \ldots, \alpha_n$ are optional temporal annotations, and $C_1, \ldots, C_k$ are constraints.

MuTACLP programs can be combined by means of two meta-level operators, i.e., union $\cup$ and intersection $\cap$. Formally, the operators define the following language of *program expressions*:

$$\mathit{Exp} ::= \mathit{Pname} \mid \mathit{Exp} \cup \mathit{Exp} \mid \mathit{Exp} \cap \mathit{Exp}$$

where *Pname* is the syntactic category of program names, each uniquely identifying a MuTACLP program.

Intuitively, names in *Pname* identify programs which are used as basic building blocks of the system. The union and intersection operators mirror two forms of cooperation between programs. In the case of union, either program may be used to derive, hence the union of two programs corresponds to put together the clauses belonging to each program. In the case of intersection, both programs must agree at each derivation step. More precisely, intersection allows one to combine knowledge by merging clauses with unifiable heads into clauses having the conjunction of the bodies of the original clauses as body, and the unified head annotated with the greatest lower bound of the head annotations as head.

MuTACLP is given an operational semantics by means of a meta-interpreter. The meta-interpreter is obtained by enriching the well-known vanilla meta-interpreter for logic programs in order to deal with the annotations and to give meaning to the composition operations. It defines the two-argument predicate *demo* which represents provability, in other words, $demo(\mathcal{E}, G)$ means that the formula G is provable in the program expression $\mathcal{E}$. For a formal definition of the semantics we refer the reader to [1].

Spatial Representation and Spatio-Temporal Correlations. Spatial information can be represented and handled inside our framework by means of constraints. A spatial object is modelled by a predicate and its spatial extent is expressed by adding variables denoting the spatial coordinates as arguments, and by placing constraints on such variables. For instance, a convex polygon, that can be seen as the intersection of a set of half-planes, is represented by a conjunction of inequalities each defining a single half-plane. A non-convex polygon, instead, is modelled as the union of a set of convex polygons.

Furthermore, the facilities to handle time offered by the language allow one to easily establish spatio-temporal correlations, like time-varying areas, or, more generally, moving objects, supporting either discrete or continuous changes. For instance, a moving point can be modelled by using a clause of the form:

$$moving\_point(X, Y)\ \mathtt{at}\ T : -constraint(X,\, Y,\, T)$$

where $constraint(X,\, Y,\, T)$ is a conjunction of constraints involving spatial and temporal variables.

In a similar way we can represent regions which move continuously in the plane. For instance, consider the area flooded by the tide, and assume that the front end of the tide is a linear function of time (the example is taken from [7]). This time-varying area can be described as

$$floodedarea(X,\, Y)\ \mathtt{at}\ T\ \ : -\ 1 \leq Y, Y \leq 10, 3 \leq X, X \leq 10, Y \geq X + 8 - T$$

4.2 The case study

In this section we apply the MuTACLP approach to the case study illustrated in Section 2.

In our experiment we focus on two problems: finding out the estimated position of the den whenever its real location is unknown and discovering which animals are likely to be a couple. In order to achieve this aim we only need a subset of the whole database and we represent the information of interest for the crested porcupine as a collection of facts of the kind:

```
fix(Id,X,Y,Hour) at Date.
```

These specify the spatio-temporal localisation of the animal `Id` giving the position `X,Y`, and the time, `Hour` and `Date`, of the bearing.

Let us now formalise the questions we want to deal with. Expert knowledge tells us that the crested porcupine stays in its den during the day whereas it usually spends the night far from it. Hence, in order to determine the position of the den we collect all the fixes of the animal which range from one hour before dawn to one hour after sunset, since in this time period the animal is probably close to its den. In order to determine whether two crested porcupines are a couple we exploit the notion of contemporary fixes: two fixes are contemporary if they are within a certain distance and in a certain time interval. The actual thresholds have to be chosen by the domain experts. Then, given a pair of animals, we estimate the number of contemporary fixes for such a pair and, according to this number, the expert can decide whether the pair is a couple or not. Notice that these are typical spatio-temporal queries in which we select spatial data depending on temporal information.

In the spirit of the MuTACLP framework, we partition knowledge among different programs. We define four programs: ANALYSIS collects the rules that implement our analysis criteria; SUN provides the predicates that allow us to determine the dawn and the sunset in the period of interest; DATAP contains the data of the bearings; finally AUX gathers together the definitions of auxiliary predicates used in the computation. The analysis rules are separated from the specific data and thus the program ANALYSIS can be reused to perform reasoning on different data sets.

The rules are slightly simplified by omitting some implementation details. This presentation allows us to focus on the knowledge representation ability of the language. The rules use the Prolog meta-predicate `findall(X,G,L)` which computes the list `L` of elements `X` that satisfy the goal `G` and in the Prolog code the symbols ∪ and ∩ denoting union and intersection of program expressions are replaced by `+` and `*` respectively.

ANALYSIS:

```
possible_loc(Id,Lloc) at T :-
    findall(loc(X,Y),demo(dataP+sun+aux,(fix(Id,X,Y,Hour) at T,
            dawn_sunset(Hour) at T)), Lloc).
prob_den(Id,Rad,Prob,L) at T :- possible_loc(Id,Lloc) at T,
                                neighbour_list(Lloc,Rad,Prob,L).

fixes_in_day(Id1,Id2,R,S,N) at T :-
    findall(c(Id1,Id2), demo(dataP+analysis+aux,(fix(Id1,X1,Y1,H1) at T,
            fix(Id2,X2,Y2,H2) at T, contem(X1,Y1,X2,Y2,R,S,H1,H2))), L),
    length(L,N).

contem(X1,Y1,X2,Y2,Rad,Sec,H1,H2):- dist(X1,Y1,X2,Y2,D),
                                    D < Rad, abs(H1-H2) < Sec.

couple(Id1,Id2,R,S,Ratio) at T :- sex(Id1,S1), sex(Id2,S2), S1!=S2,
                                  fixes_in_day(Id1,Id2,R,S,N) at T,
                                  fixes_in_day(Id1,Id2,1000000,S,M) at T,
                                  Ratio < (N/M).
```

The first and second clauses are used for the den analysis. The first one returns the list of positions `Lloc` of the animal `Id` between dawn and sunset in a given day, i.e., the positions in the fixes whose hour of bearing falls within the part of the day of interest. Note that this clause uses the predicate `demo` defined by the meta-interpreter of MuTACLP: the first argument of `demo` specifies the program expression where a goal has to be solved while the second one is the goal to be solved. Hence, in this case the conjunctive atom `fix(Id,X,Y,Hour) at T, dawn_sunset(Hour) at T` is computed in the *union* of the programs `dataP`, `sun` and `aux`. The second clause extracts from the list of positions `Lloc` those which are probable dens. The predicate `neighbour_list`, later described more in detail, selects those positions whose neighbourhood (with radius `Rad`) includes a great quantity of fixes between dawn and sunset. This quantity is estimated by considering the ratio between the number of fixes in the neighbourhood and all the fixes for the given animal in the period of time of interest.

The remaining clauses are aimed at finding the (possible) couples. The predicate `fixes_in_day` returns the number `N` of contemporary fixes in a day for the pair of crested porcupines `Id1,Id2`. Two fixes are judged contemporary if their spatial and temporal distance is bounded by `R` and `S` respectively. The predicate `contem` checks whether two fixes are contemporary. Finally, the predicate `couple` checks whether two individuals `Id1` and `Id2` should be considered a couple. This is done by verifying if the ratio between the number of contemporary fixes of the crested porcupines `Id1,Id2` and the number of observations of `Id1,Id2` within `S` seconds at arbitrary distance (in practice this is obtained by setting a very large bound for the distance parameter) in a certain day overcomes a given threshold.

The predicates `neighbour_list`, `dist` and `length` are defined in the program AUX. The predicate `neighbour_list` returns a list of `pden(X,Y,N)` specifying the position `X,Y` of a probable den and the number `N` of fixes (between dawn and sunset) within its neighbourhood, whereas `dist` and `length` compute the distance `D` between two points and the length of a list, respectively.

We complete the presentation by showing the code for the other programs.

SUN:

```
dawn_sunset(Hour) at T :- light(D,S) at T,
                           between_ds(D,S,Hour).
between_ds(D,S,Hour):- Before_dawn is D-3600,
                       After_sunset is S+3600,
                       Hour>=Before_dawn,Hour=<After_sunset.
light(25470,63910) th [[1,1,1998],[31,1,1998]].
⋮
light(25530,62820) th [[1,12,1999],[31,12,1999]].
```

DATAP:

```
 fix(f1,62060.0,1669490.0,4724115.0) at [01,01,1998].
 fix(f3,62120.0,1669740.0,4724100.0) at [01,01,1998].
 fix(f2,76380.0,1669535.0,4724390.0) at [01,01,1998].
 ⋮
```

AUX:

```
 neighbour_list(Lloc,Rad,Prob,L):-
                              neighbour(Lloc,Lloc,Rad,Ltemp),
                              length(Ltemp,N),
                              select(Ltemp,N,Prob,L).
⋮
```

In the program SUN the predicate `dawn_sunset` checks whether the hour of the bearing is between one hour (3600 seconds) before dawn and one hour after sunset. Notice that, since dawn and sunset times vary throughout the year, the predicate `light` records a monthly estimate of such data expressed in seconds. The program DATAP collects the data on the crested porcupines provided by the behavioural ecologists: they consist in 4000 fixes concerning 25 different individuals. Finally the program AUX collects the definitions of six auxiliary predicates used in the computation. Only one of these definitions is shown explicitly.

It is worth noting that time is expressed by dates instead of real numbers. An obvious translation converts dates into numbers during the computation.

The problems concerning crested porcupines mentioned at the beginning of this section, i.e., couple identification and den localisation can now be tackled by ecologists by appropriately querying the system. For instance, suppose we want to know whether the crested porcupines identified as m15 and f3 are a couple in some days of the time period from 26 December 1998 to 29 July 1999. According to the suggestions of experts, we set the thresholds under which fixes are considered contemporary parameters to 100 meters and to 15 minutes (900 seconds).

```
| ?- demo(analysis+aux,couple(m15,f3,100,900,0.5) in [[26,12,1998],[29,7,1999]]).
```

The answer to the query is yes, hence we can conclude that the animals m15 and f3 are probably a couple.

To confirm this hypothesis, behavioural ecologists need to know whether they have the same den in that period of time.

```
| ?- demo(analysis+aux,prob_den(m15,100,0.45,L) in [[26,12,1998],[29,7,1999]]).

L = [pden(1669880.0,4724010.0,17), pden(1669900.0,4724020.0,17),
    pden(1669895.0,4724025.0,17),pden(1669855.0,4724040.0,17)]

| ?- demo(analysis+aux, prob_den(f3,100,0.47,L) in [[26,12,1998],[29,7,1999]]).

L = [pden(1669920.0,4723990.0,21),pden(1669920.0,4723990.0,21),
    pden(1669870.0,4724000.0,21), pden(1669910.0,4724030.0,21),
    pden(1669920.0,4724000.0,21), pden(1669900.0,4724020.0,21),
    pden(1669915.0,4724030.0,21),pden(1669900.0,4724060.0,21)]
```

Each of the above queries returns a set of probable den locations for the given crested porcupine. According to the answers obtained, the two animals have their dens located in an area of 60 meters diameter. As stated in Section 2.2, the radio-tracking technique presents several sources of error and behavioural ecologists estimated an error of about 62 meters in the given data. Thus the answer is coherent with the previous deduction that m15 and f3 are a couple. The results inferred using the system have also been validated by the behavioural ecologists comparing them with their data. For example, the *homing-in* data reveal that in the considered time period the two animals `m15` and `f3` have a common den located at `X = 1669960, Y = 4724040`, whose distance from the computed probable dens is around 50-80 meters.

4.3 Summary Assessment

The case study highlights some advantages of MuTACLP. First of all, having a declarative language allows one to represent in a natural and powerful way the expert knowledge. For instance, the code defining the predicates `prob_den` and `couple` expresses in a compact way the criteria suggested by the behavioural ecologists to discover dens and couples. It is important to remark that we do not represent only data (e.g., the `fix` predicate) as in constraint databases [13, 3] but also rules (e.g., the `prob_den`, `couple` predicates). This extra feature makes the difference when the formalism is used as specification and/or analysis language.

In addition, MuTACLP supplies a deductive engine thanks to which we can offer high-level mechanisms to reason and extract new information from data. Such an engine automatically takes into account temporal information thus supporting temporal reasoning. For instance, we can easily ask the system for the location of the den of an animal within a given period of time. The den is not a stored information but a derived one, obtained by considering spatio-temporal constraints.

The pieces of temporal information are given by temporal annotations which says at what time(s) the formula to which they are attached is valid. Annotations make time explicit, but avoid the proliferation of temporal variables and quantifiers of the first-order approach. For instance, the `th` annotation is used in the SUN program to express that *for every* day of a certain month, the hours of dawn and sunset assume determinate values. Instead, in the query about couples we use the `in` annotation because we want to know whether there *exists* a time point within a given interval which satisfies the couple condition.

Another feature of MuTACLP which has been proved useful in the modelling of the case study is modularity: the knowledge has been partitioned into 4 modules, later combined using the composition operator `+` in order to solve the queries of interest. Few logic-based approaches supply the user with modularity features. Among them, we recall the multi-theory framework proposed by Subrahmanian [32]. It is based on *annotated logics* and it is a very general framework aimed at amalgamating multiple knowledge bases. Temporal information can be handled whereas no support for spatial data is given.

To adequately appreciate our logical framework, we refer to next section in which MuTACLP is employed on top of a GIS in order to provide a more friendly interface for GIS analysis.

5 ArcView: the GIS point of view

In order to better understand the opportunities and the limitations of the current GIS technology on spatio-temporal reasoning and to look for possible integrations of such technology with our logical framework, we tried to produce a system which satisfies the requirements listed in Section 2.3, by using ArcView [10] by ESRI. The aim of this system is to provide the behavioural ecologist with a tool that, on the one hand, automates, as much as possible, the basic procedures that are usually done by hand, and, on the other hand, offers a set of spatio-temporal analysis functions which are typically not supported by standard GIS. The choice of ArcView as the GIS Software is due to its importance and diffusion. ArcView is a desktop solution offered by ESRI and provides data visualisation, query, analysis, and integration capabilities along with the ability to create and edit geographic data. We used two different versions of this product, ArcView 3.2 and ArcView 8.1. Building on the experience with ArcView 3.2, in the second case, we have developed a system which integrates ArcView 8.1 with the potentiality of MuTACLP.

5.1 Implementation in ArcView 3.2

The spatio-temporal porcupine application developed with ArcView 3.2 exploits the extension Spatial Analyst (ESRI) which provides raster capabilities. It also exploits the open source package Movement [26] which, relying on the Spatial Analyst functionalities, implements a set of functions, like Home Range estimators, in order to analyse the movement and behaviour of animals. Based on this software our application has been built using Avenue. This is a procedural script programming language offered by ArcView 3.2 for customisation purposes. The description of the developed system can be found in [6]. From our experience, due to its declarative nature, the code for MuTACLP is much more compact and readable than the corresponding script written in Avenue. On the other hand, commercial GISs offer many visualisation and spatial handling mechanisms that are not available in MuTACLP.

5.2 Implementation in ArcView 8.1

The experience illustrated in the previous section led us to the idea of designing a framework which offers the user, at the same time, the efficient functionalities, the graphical user interface, and the visualisation capabilities of a commercial GIS, and the high-level declarative representation mechanisms and query language for spatio-temporal data of MuTACLP. More precisely, we have developed a new version of the system which exploits ArcView 8.1 (a more recent release of ArcView) and integrates the GIS with a logical component implementing the deductive language MuTACLP. The resulting framework should ideally combine the advantages of both systems while reducing their drawbacks.

ArcGIS 8.1 is a completely new product with respect to the previous release. It provides an integrated environment that includes ArcView, ArcInfo, ArcSDE and ArcIMS. One of the main features of this new GIS environment is that the programming technology is based on Common Object Model (COM). This means that ArcGIS supports any language based on COM, such as Visual Basic or Visual C++.

The architecture of the developed system [12, 27] is illustrated in Fig. 9. It presents two main components: the GIS Module and the Logical Module. The GIS Module provides the geographical data handling, the storing capabilities (by means of ArcView) and the TCP/IP connection to the Logical Module (using Visual Basic). The Logical Module relies on Sicstus Prolog which, besides being used for implementing MuTACLP and the domain knowledge, represents the inference engine for the integrated system.

The user interacts directly with the GIS Module. He/she can perform analysis on animal behaviour using:

- STIA (Spatio-Temporal Individual Analysis), a customisation of the standard GIS functionalities, or
- enhanced spatio-temporal capabilities provided by the logical module.

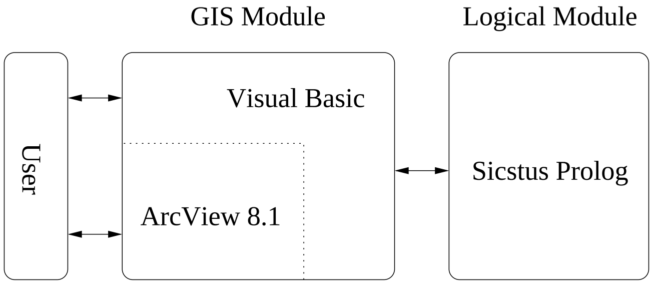


Fig. 9. The integrated architecture

Fig. 10 shows the interface through which the user can access the system. Such an interface is the typical ArcView interface enriched with new menus which allow one to access STIA functionalities (on the right hand side of the upper tool bar) and to exploit the additional spatio-temporal logic module (the “Call Sicstus Prolog” button on the left hand side of the upper tool bar).

The functionalities provided by STIA, whose menu tool bar is shown in Fig. 11, are grouped into three categories:

- tools for analysing relationships among individuals,
- tools for studying the behaviour of individuals with respect to events, and
- tools for configurating the system.

The aim of the first group of functions is to study the behaviour of pairs of animals. As already discussed in Sections 2.3 and 4.2, the solution for problems of this kind is based on the notion of contemporary fixes. Here fixes are first grouped according to their temporal component, and only later, for some kinds of queries, also the spatial component is managed. Say that two fixes are considered *simultaneous* if they fall within a time interval, called zero-interval, whose duration and scale (seconds, minutes, days) are input parameters, which can be fixed by the user depending on

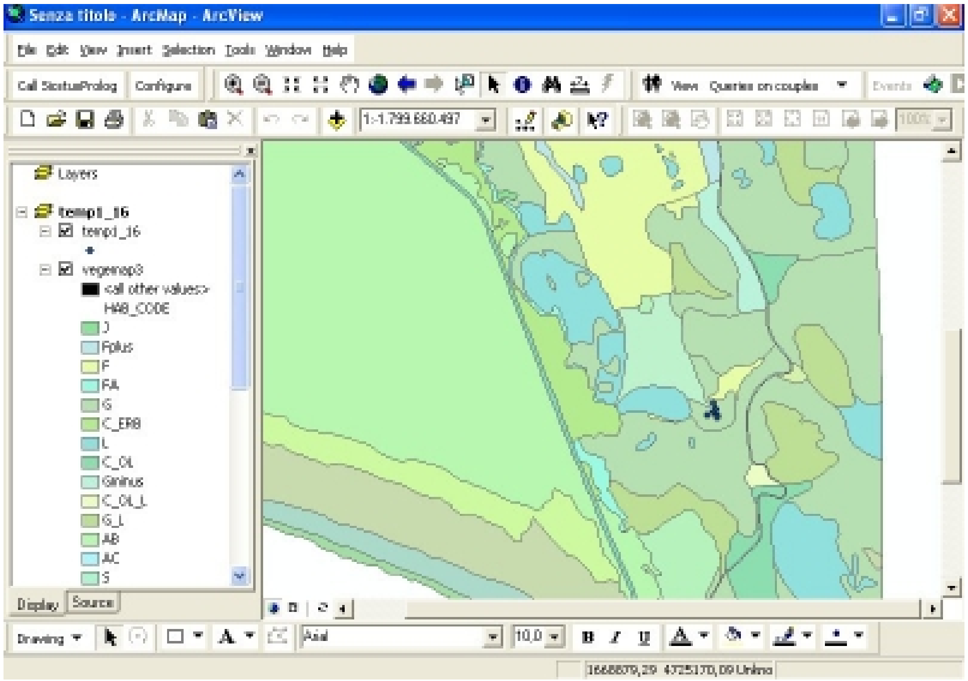


Fig. 10. Screenshot of the integrated system

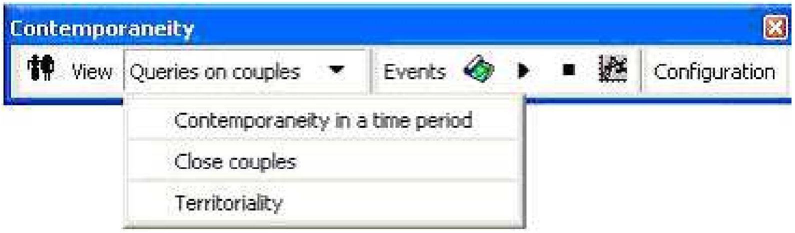


Fig. 11. The menu tool bar of STIA

the context. Then, the button 👥 produces a table containing all the pairs of simultaneous fixes, each associated with the spatial and temporal distance between the involved fixes. This table already represents a valuable source of information for behavioural ecologists. Moreover, it is a basic data structure for the implementation of all the other functionalities in the same class, i.e.:

- **View and Contemporaneity in a time period**. Such functions allow the user to choose among different options of visualisations. One can select to view the positions and the relative distance of a given pair of animals, or of all the pairs in which a given animal is present. Moreover, one can set the time period of interest.

- **Close couples**. This function selects and visualises the contemporary fixes as specified in Section 4.2, in a given period of time. Such a function allows one to infer the couples.
- **Territoriality**. This function detects animals that avoid other animals. It is based on the assumption that an animal avoids another one if their inter-individual distance is always greater than a minimal fixed value.

The second group of functionalities allows the user to analyse the behaviour of animals with respect to a certain event. We proceed as above: first a table containing all the information about fixes and events is created, and then such a table is used to visualise and implement the queries of interest. The table is obtained as the Cartesian product between the table of fixes and the table of events. For each entry it contains also the spatial and temporal distances between the event and the fix. The main functionality provides the possibility to select an event, an animal and to set some spatial and temporal constraint on the influence of events over individuals. As result, the system displays an animation that shows how animals move with respect to the selected event, thus making it easier for the user to have a preliminary idea of the animal's reaction to the chosen event. Fig. 12 illustrates a result of this query. In this case the event is the snow and the area where it happens is delimited by a circle (right upper corner of the map), whereas the locations of the individuals are represented as points associated with their identifier and the time instant of the bearing.

Finally, the configuration section allows the user to customise the system in order to analyse the behaviour of different kinds of animals detected by a radio-tracking technique. For example, the user can choose which are the sources of data to be analysed, where to store the results, the time granularity and the width of the zero-interval, and in which kind of raster map to visualise the query results (e.g., in Fig. 12 the map shows the vegetation and the cultivated areas).

It is worth noting that the implementation of STIA greatly exploits the features of Visual Basic, such as a better integration with MS Windows objects, the use of ArcObjects library provided by ESRI, the use of MS Access for storing spatial objects (geodatabase) and therefore this provides the application developer with an almost standard SQL language.

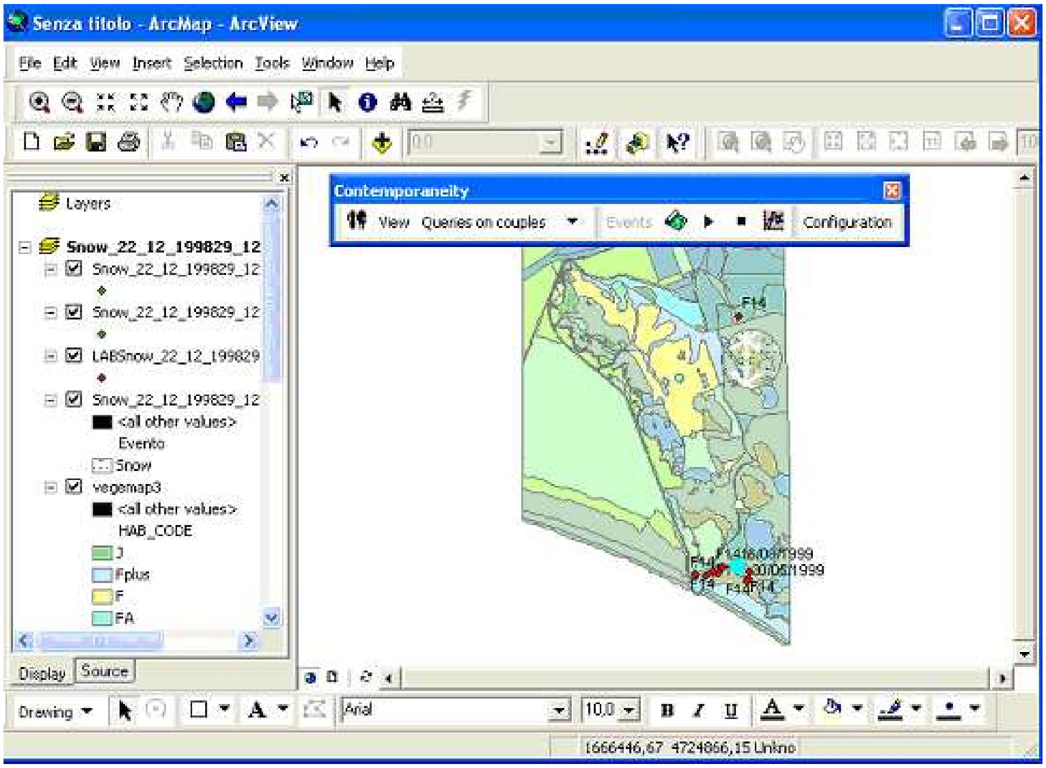


Fig. 12. Screenshot of a query on the relationship between animals and events.

We finally give some hints on the integration with the Logic Module that enriches the system with a spatio-temporal inference engine. Such an integration has been carried out by means of a TCP/IP connection between the Visual Basic part and Sicstus Prolog part [29], where MuTACLP is implemented. Each command, written in an ad-hoc interface, is sent by means of TCP/IP sockets from the ArcGIS user interface to the MuTACLP meta-interpreter and finally to the Prolog engine that performs the computation. Whenever an operation on spatial objects is asked for (e.g., the addition of an object to a layer), the request is sent to the GIS that executes such an operation. ArcView returns the identifier of the resulting object to the Prolog interpreter, which continues the deduction process. When a solution is found, it is sent back to the GIS for visualisation (for more implementation details see [19]).

To activate the connection with the Logic Module the button *call Sicstus Prolog* (on the left of the tool bar in Fig. 10) is used. For example, Fig. 10 shows the results of the computation of the den discovery. This computation exploits the Sicstus Prolog code described in Section 4.2, combined with the visualisation functions of ArcView. The possible dens are represented by points and the location inside the map improves the analysis, e.g., the user can immediately identify possible relationships with characteristics of the area, like the kind of ground or vegetation. Moreover, we can easily perform further spatial analyses by overlapping different layers obtained as results of various queries. For instance, as stated in Section 4.2, to confirm the fact that two crested porcupines form a couple one can compare the den position. This can be performed by combining the results of the den queries in a unique map, thus allowing the user to ascertain easily whether the dens of the animals are close to each other.

6 Conclusions

In this paper, we have described how some recently developed ideas in the area of data modelling and reasoning can be profitably used to construct software which supports a specific application area - the study of animal behaviour. The measure of our success is given by the positive reactions we have had from the behavioural ecologists, and by the demonstration that certain behavioural analyses, requiring a deep understanding of behavioural rules and of the structure of a territory, can automatically be supported. So far, in this framework, software tools have been employed only for either standard statistical analysis or for data storage and retrieval.

In this respect the application of MADS and MuTACLP has proven very useful at the levels of conceptual modelling and implementation of complex spatio-temporal data extraction procedures which extend the capabilities of currently available software.

The use of the object-oriented, extended notation in MADS has captured a number of spatio-temporal dimensions implicit in the selected queries, otherwise very difficult to represent in standard relational data models. The rich spatial notations, the explicit temporal notations and the aggregation links have allowed entities, like individuals, to be modelled in their association with each other (i.e. as member of a family) and with the environment (their habitat and home range, as well as the occurrence of specific events impacting on their behaviour). This has been fundamental in the considered application area.

The use of MuTACLP has taken such a conceptual modelling to the subsequent stage of implementing the selected spatio-temporal queries. Concerning the first problem (den localisation) the MuTACLP query allows the user to identify fixes likely to be a den. As concerns the second problem (association among animals) the application developed in MuTACLP calculates inter-individual distance in space and time, identifying cases of association, like couples, and it can also be used to determine which animals avoid others. Apart from the added advantage of the level of abstraction for programming the queries offered by MuTACLP, such an automation cannot be obtained in a simple way by using a standard GIS.

Such results have convinced us of the importance of pushing forward our efforts both with respect to the development of the supporting technologies and the search for new application fields. With respect to the technologies, we are now integrating data mining and knowledge discovery technology in the framework. We are developing new algorithms for clustering trajectories and for classifying them. It is evident that such tools can help in implementing new queries of interest in behavioural ecology. As to the application fields, the large availability of these forms of geographic information is expected to enable novel classes of applications, where the discovery of consumable, concise, and applicable knowledge is the key step. For instance, very high resolution satellite (VHR) imagery will revolutionize the business of GISs, and will enable complex, dynamic, social and environmental phenomena, like urban development, to be monitored and pollution patterns to be detected. Furthermore, the emergence of high resolution satellites will produce huge amounts of data. At this moment, even a single satellite system is expected to accumulate over one terabyte of data every day. As another example, the presence of large numbers of location-aware, wirelessly connected mobile devices gives a growing possibility to access the space-time trajectories of these personal devices and their human companions. These mobile trajectories contain detailed information about personal and vehicular mobile behaviour and therefore offer interesting practical opportunities for the finding of behavioural patterns. Such information can be used for instance in traffic management.

Acknowledgements We would like to thank Dr. A. Sforzi of the Ethology and Behavioural Ecology Group (University of Siena) and A. Brandini, N. Grasso and A. Isolani who collaborated in the implementation of the systems. We also thank P. Baldan and E. Fullwood for their careful reading of the paper. We are grateful to the anonymous reviewers for their insightful comments. This work has been supported by Esprit Working group 28115 DeduGIS and partially by the MIUR Italian Project GeoPKDD.

References

1. P. Baldan, P. Mancarella, A. Raffaetà, and F. Turini. MuTACLP: A language for temporal reasoning with multiple theories. In *Computational Logic: Logic Programming and Beyond*, volume 2408 of *LNAI*, pages 1–40. Springer, 2002.
2. Y. Bédard, S. Larrivée, M. Proulx, and M. Nadeau. Modeling Geospatial Databases with Plug-Ins for Visual Languages: A Pragmatic Approach and the Impacts of 16 Years of Research and Experimentations on Perceptory. In *ER (Workshops)*, volume 3289 of *LNCS*, pages 17–30. Springer, 2004.
3. A. Belussi, E. Bertino, and B. Catania. An extended algebra for constraint databases. *IEEE TKDE*, 10(5):686–705, 1998.
4. W. G. Berendsohn, A. Anagnostopoulos, J. Jakupovic, P. L. Nimis, and B. Valdés. A Framework for Biological Information Models. In *Proceedings of the VIII OPTIMA meeting*, volume 19, pages 667–672, 1996.
5. M.H. Böhlen, C.S. Jensen, and M.O. Scholl, editors. *Spatio-Temporal Database Management*, volume 1678 of *LNCS*. Springer, 1999.
6. A. Brandini. Analisi Spazio-Temporale con GIS commerciali per lo studio del comportamento animale, 2001. Tesi di Laurea, Università degli Studi di Pisa.
7. J. Chomicki and P.Z. Revesz. Constraint-Based Interoperability of Spatiotemporal Databases. *GeoInformatica*, 3(3):211–243, 1999.

8. M.T. Corsini, S. Lovari, and S. Sonnino. Temporal activity pattern of cresteed porcupines hystrix cristata. *Journal of Zoology*, 256:43–54, 1995.

9. R. Deitner and K. Boykin. An Entity-relationship Model of Wildlife Habitat Associations. In *Special Session for the 19th Annual Unites States Regional Association of the International Association for Landscape Ecology*, 2004. Las Vegas, NV.

10. ESRI - Environmental Systems Research Institute, http://www.esri.com.

11. A. Felicioli and L. Santini. Burrow entrance hole orientation and first emergence time in the crested porcupine hystrix cristata l.: space-time dependence on sunset. *Polish Ecological Studies*, 3/4(20):317–321, 1994.

12. N. Grasso and A. Isolani. Un sistema che integra basi di dati geografiche e ragionamento spazio temporale, 2002. Tesi di Laurea, Università degli Studi di Pisa.

13. S. Grumbach, P. Rigaux, and L. Segoufin. Spatio-Temporal Data Handling with Constraints. *GeoInformatica*, 5(1):95–115, 2001.

14. R. Kenward. *Wildlife radiotagging.* Academic Press, 1987.

15. V. Khatri, S. Ram, and R. T. Snodgrass. Augmenting a Conceptual Model with Geospatiotemporal Annotations. *IEEE TKDE*, 16(11):1324–1338, 2004.

16. M. Koubarakis and S. Skiadopoulos. Tractable Query Answering in Indefinite Constraint Databases: Basic Results and Applications to Querying Spatiotemporal Information. In *[5]*, pages 204–223, 1999.

17. G. M. Kuper, L. Libkin, and J. Paredaens. *Constraint Databases.* Springer, 2000.

18. M. Lucherini. Attività e uso dello spazio nell'istrice hystrix cristata, 1996. Tesi di Dottorato di Ricerca, Università degli Studi di Siena.

19. P. Mancarella, A. Raffaetà, C. Renso, and F. Turini. Integrating Knowledge Representation and Reasoning in Geographical Information Systems. *Journal of GIS*, 18(4):417–446, 2004.

20. P. Mancarella, A. Raffaetà, and F. Turini. Temporal Annotated Constraint Logic Programming with Multiple Theories. In *[31]*, pages 501–508. IEEE Computer Society Press, 1999.

21. C. Parent, S. Spaccapietra, and E. Zimanyi. Spatio-temporal conceptual models: Data structures + space + time. In *ACM GIS*, pages 26–33. ACM Press, 1999.

22. C. Parent, S. Spaccapietra, and E. Zimanyi. *Conceptual Modeling for Traditional and Spatio-Temporal Applications: The MADS Approach.* Springer Verlag, 2005. To appear.

23. C. Parent, S. Spaccapietra, and E. Zimanyi. The MurMur Project: Modeling and Querying Multi-Representation Spatio-Temporal Databases. *Information Systems*, 2005. To appear.

24. D. J. Peuquet. Making Space for Time: Issues in Space-Time Data Representation. *GeoInformatica*, 5(1):11–32, 2001.

25. G. Pigozzi. Crested porcupines hystrix cristata within badger setts meles in the maremma natural park, italy. *Säugetierkundliche Mitteilungen Band*, 33(2/3):261–263, 1986.

26. W.M. Eichenlaub P.N. Hooge and E.K. Solomon. Animal movement extension to arcview. ver. 2.0. Alaska Biological Science Center, U.S. Geological Survey, Anchorage, AK, USA. 1997.

27. A. Raffaetà, C. Renso, and F. Turini. Enhancing GISs for Spatio-Temporal Reasoning. In *ACM GIS*, pages 35–41. ACM Press, 2002.

28. L. Santini. The habits and influence on the environment of the old world porcupine Hystrix cristata L. in the northernmost of its range. In *Atti del Congresso Ninth Vertebrate pest conference, Fresno, California*, pages 149–153, 1980.

29. SICS. *Sicstus Prolog User's Guide*, 1995.

30. S. Sonnino. Spatial activity and habitat use of crested porcupine, hystrix cristata l. 1758 (rodentia, hystricidae) in central italy. *Mammalia*, 2(62):175–189, 1998.

31. S. Spaccapietra, editor. *Spatio-Temporal Data Models & Languages (DEXA Workshop).* IEEE Computer Society Press, 1999.

32. V. S. Subrahmanian. Amalgamating Knowledge Bases. *ACM Transactions on Database Systems*, 19(2):291–331, June 1994.

33. N. Tryfona, R. Price, and C. S. Jensen. Conceptual Models for Spatio-temporal Applications. In *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, volume 2520 of *LNCS*, pages 79–116. Springer, 2003.

34. G.C. White and R.A. Garrott. *Analysis of wildlife Radio-Tracking data.* Academic Press, 1990.

35. B.J. Worton. Kernel methods for estimating the utilization distribution in home range studies. *Ecology*, 1(70):164–168, 1989.