Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

UTILIZING SPATIO-TEMPORAL DATA IN MULTI-AGENT SIMULATION

Daniel Glake
Norbert Ritter

Department of Informatics
University of Hamburg
Vogt-Kölln-Straße 30
22527 Hamburg, GERMANY

Thomas Clemen

Department of Computer Science
Hamburg University of Applied Sciences
Berliner Tor 7
20099 Hamburg, GERMANY

ABSTRACT

Spatio-temporal properties strongly influence a large proportion of multi-agent simulations (MAS) in their application domains. Time-dependent simulations benefit from correct and time-sensitive input data that match the current simulated time or offer the possibility to take into account previous simulation states in their modelling perspective. In this paper, we present the concepts and semantics of data-driven simulations with vector and raster data and extend them by a time dimension that applies at run-time within the simulation execution or in conjunction with the definition of MAS models. We show that the semantics consider the evolution of spatio-temporal objects with their temporal relationships between spatial entities.

1 INTRODUCTION

The significance and high utility of spatial and temporal data in decision-support are without question. Crisis management, environmental management, and other areas of application of multi-agent simulation scenarios demand feasible integration mechanism for vector and raster data objects, often in conjunction with time-series.

Even in the event of so-called Temporal GIS (TGIS) systems (Worboys 1998), generic methodological approaches for dealing with spatio-temporal data adequately are missing (Siabato et al. 2018). Our goal within the scope of this paper is to derive a functional data model for dealing with spatio-temporal data in multi-agent simulations. A significant part of that model has been implemented already within the Multi-Agent Research and Simulation framework (MARS) (Glake et al. 2017; Weyl et al. 2018), and was utilized in large-scale scenarios in Weyl et al. (2019), Berger et al. (2019) and Lenfers et al. (2018).

2 RELATED WORK

A TGIS aims to process, manage, and analyze spatio-temporal data. However, the capabilities of an information system depend primarily on the design of its data models. Data models represent the conceptual core of an information system; they define data object types, relationships, operations, and rules for maintaining database integrity. A rigorous data model must anticipate spatio-temporal queries and analytical methods for execution in the TGIS. Information about temporal structures must be represented by data objects and made available for analysis in decision-making in multi-agent simulation by suitable concepts. If a temporal GIS does not have a good data model, its support for temporal queries and temporal analysis of phenomena is ineffective.

Previous GIS data models focus on representations of reality by static information. A given geographic area is decomposed into a multiple individual layers as regular (*raster*) or irregular (*vectors*). Decoupled layers object limit the GIS in representing dynamic information, such as *transitions* and *movements*. Raster

978-1-7281-9499-8/20/$31.00 ©2020 IEEE

cells encode attribute values at any given location without regard to the spatial properties of the subject they represent. Geometrically indexed vector objects, on the other hand, force a segmentation of the entities to be represented into separate layers when they interact in time or space. GIS requires a complete and rigorous framework for modelling geographic data to overcome the difficulties of dealing with geographic complexity, scale differences, generalization, and accuracy. The lack of data representation schemes to integrate GIS data with models for spatio-temporal simulations is a significant deficit of current simulation frameworks. However, several tools have already been proposed to query such related types. The STQL (Erwig and Schneider 2002) is a query language and extension of relational SQL. Besides integrated spatial operators, the focus here is on so-called temporal lifting operators as well as temporal selections and aggregations. Temporal Lifting describes the query of an object at a given time, used within a sub-clause. In contrast, the language SQL^{ST} works by using triangles as its base and integrates the specifications for temporal GIS described by Worboys (1994). Each vector-geometry is therefore transformed into a respective triangle representation and implements multiple spatial operations. Existing simulation frameworks like NetLogo, Repast Symphony (North et al. 2013) and GAMA (Grignard et al. 2013) only support the use of spatial data by extension. For example, GAMA offers, among other features, the capability to import shapefiles and OSM files, in order to use them as the underlying simulation environment, consisting of importing the file and defining a mapping of the attributes in the feature tables to the agents, denoted as species. Repast provides a suitable GIS extension in order to perform spatial queries on the vector layer, whereas NetLogo is supporting raster inputs as well. None of them supports a temporal dimension in the input data processing. Users need to adapt their models in order to respect temporal changes in spatial objects.

3 INTRODUCTION MARS FRAMEWORK

MARS provides an ecosystem for developing multi-agent simulations, based on the Modeling and Simulation as a Service paradigm (Weyl et al. 2019). End-users can create their simulations in a variety of ways and execute them directly on their machine or in the dedicated MARS cloud (Weyl et al. 2018). The system is designed to serve in cloud-native environments, thereby scaling up the simulations and considering update-intensive state management. Results are persisted in multiple databases or files, prepared used in our visual analytics board or within a 3D visualization, introduced by Dalski et al. (2017).

This paper describes a new perspective integrated into our MARS-DSL modelling language (Glake et al. 2017). The model considers two types of state updates seen in practice as well as geometric queries restricted by temporal predicates, in which federated queries consider spatial- and temporal dimensions. We make the following contributions:

- (Section 4) We provide a detailed analysis of current challenges, considering temporal changes for the spatial objects under the circumstances for multi-agent simulations.
- (Section 4.3) Towards a meaningful data integration of spatio-temporal data, we define the conceptual model formally as an *abstract data type* and required *operations*.
- (Section 5.3) Spatial data management in the form of geographic layer is a well-known approach handling *raster*- and *vector*-based data. In this respect, we propose *temporal-raster* and *temporal-vector* layer approach, utilizing incremental updates and extending each data type by a temporal dimension. Considering the implementation of models, we describe the integration of our approach within our existing MARS DSL language.

4 SPATIO-TEMPORAL CHALLENGES

Changes of entity- and environment attributes are essential in order to become desirable emergent phenomenons, environmental updates, and changes in process mechanisms or just within the agent's behaviour logic (Xie et al. 2016). There are six major types of spatial or temporal changes which need to be considered by a geographic-driven multi-agent simulation (Zhang et al. 2017):

Type 1. Actions over space Given a fixed time-point (e.g., a concrete simulation step) in which a certain phenomenon occurs (e.g., environmental conditions as by weather) and may change the characteristics of data objects over multiple locations or even globally, analysis is done by fixing time, controlling attributes, and measuring location.

Type 2. Messages over space Messages as like interactions between agents may contain their location over time, analysis of them is done by fixing attribute set, measuring location and controlling time.

Type 3. Attributes over time Given a period in which *attribute values* (e.g., of the agent entities) can change over time and space, analysis is done by fixing time, controlling locations, and measuring attributes.

Type 4. Actions over time Multiple actions can happen for a fixed given location (e.g., read and write actions for agents) and may change over time, analysis in the model are done by fixing location, controlling attributes, and the measuring time.

Type 5. Messages over time Given event where its characteristics or processes may change at sites through time, analysis is done by fixing attributes, controlling locations, and measuring time.

Type 6. Area over space and time Given an area where attributes may change over time and changes over space, analysis is done by fixing the attribute set, measuring time-period and restricting the environment.

The value changes in *actions over time* can be stored using a traditional event source, transactional log or any history mechanism. While changes in *attribute over time* and *area over space time* are made *directly* to associated attributes or geometric structures, the action *actions over space* considers changes in *static* spatial information. Especially densely distributed information such as substance measurements and interpolations, e.g., the temperature measurements or the distribution of substances in water. While such temporal courses in geography are mostly represented by heat and isarithmic (Ratti and Richens 2004), chloropleth (Tennekes 2018) or dasymetric maps (Jia and Gaughan 2016) for simplified processing, in previous GIS systems these vector- and raster-maps are switched on of off, depending on the time interval (Siabato et al. 2018).

4.1 Temporal Changes

Temporal changes occur at different points in time or periods and are recognized by the fact that spatial properties or the location are changing. Two types of temporal changes have to be distinguished for multi-agent simulations, the movement of objects, and the evolution of them over time. Evolution refers to changes that occur as a result of events or agent processes or their interaction with their environment (Siabato et al. 2018). Figure 1 shows *Storms* which are often related to a limited location in the simulated world and can be broken down into individual processes such as *rainfall*, *strong winds*, and *hail*. These individual actions affect subsets of spatial entities and have a transitive influence through their interrelationships (Zhang et al. 2017). For example, simulated precipitation in a particular region causes a changed *water level* in which affections lead to flooding or low tide (Mehrotra and Sharma 2009). All action-changes are decomposed over different points in time when they occur. Related to the TGIS model, *area over space time* is the development of a type of processes or events, semantically objects by two sets of temporal objects. Comparisons are made between the two sets of spatial and temporal objects to show how a process evolves its attributes, temporal properties, and spatial characteristics in the two sets of time series (Bettini et al. 2000).

Movement concerns the journey of an event or entity from one location to another. The event or entity may or may not be associated with spatial characteristics other than the location. For example, the movements within a city-simulation, in which drivers move along a pre-defined route. The changes denote a single event or process. They can be represented by linking a semantic object to a set of temporal objects and then to a set of spatial objects to show the movement of that event during the period consisting

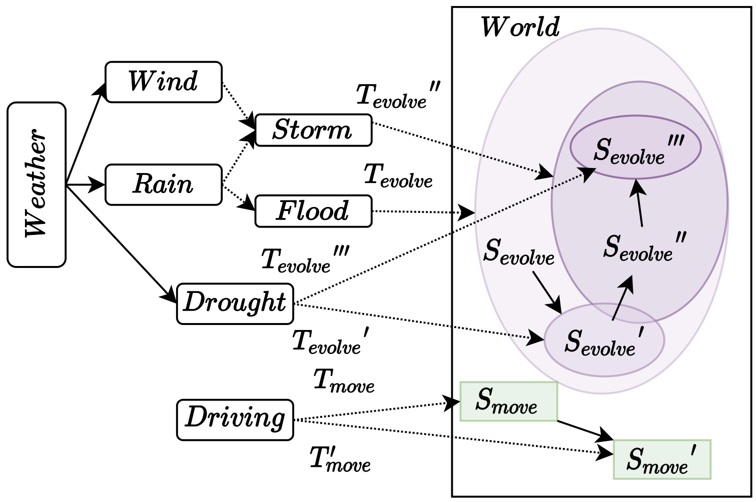

Figure 1: Example temporal change with *evolution* 4 and *move* 4 of spatial objects over time from one location to another.

of those temporal objects. Multiple subtypes for temporal changes result from combining the attribute-, morphology- and topology-change (Shekhar et al. 2015).

4.2 Spatial Changes

Spatial changes refer to spatial variations at a certain point in time or in a period in which comparisons between two or more sites are made using data from the same vintage (Worboys 1998). These changes can be classified as *static* or *transitional*, in which the static changes concerns variations of geographical influences for a particular snapshot. In contrast, transitional changes do compare modifications of actions or processes at different locations, e.g., the ecological effects of low tides and flood on a static area, which will not change its location but their water level (Siabato et al. 2018).

Transitional changes describe the variations of spatial properties for a given process or action (Zhang et al. 2017). These changes can be represented by links from temporal objects to spatial objects, in which three parameters are responsible for measuring the spatial changes: *duration*, *continuity*, and their *attributes*. For instance, figure 2 examines the urbanization process of a town in which temporary changes occur by extending the location from one time-point to another.

4.3 Conceptual Modeling

States and changes of these states are one of the essential aspects of spatio-temporal data management for the recognition of causal relationships between processes that sometimes have different effects. The aim is to answer appropriate location- and time-dependent questions with the given TGIS and to draw result-oriented conclusions (for example, the effects of nearby construction measures in coastal areas). For simulations, three basic questions arise *What happened in an area? What has happened to the area? How do we know that something has happened?* According to Worboys (1998), three components can be identified, *states*, *processes* or *actions* and *evidence*: States describe the distribution of objects either represented as a snapshot for a geometric object in time or in a space-time composite with included changes. Processes and actions cause changes in the entities themselves over time (e.g., in the case of urbanization of a city). They occur not arbitrary but are evidenced and observed, e.g., it must be ensured that we identify changes as such and that they can be recorded for the simulation as a result.

Furthermore, TGIS data models have to take into account the behaviour of phenomena. Until now, the physical processes of natural phenomena have been neglected in the development of conventional GIS

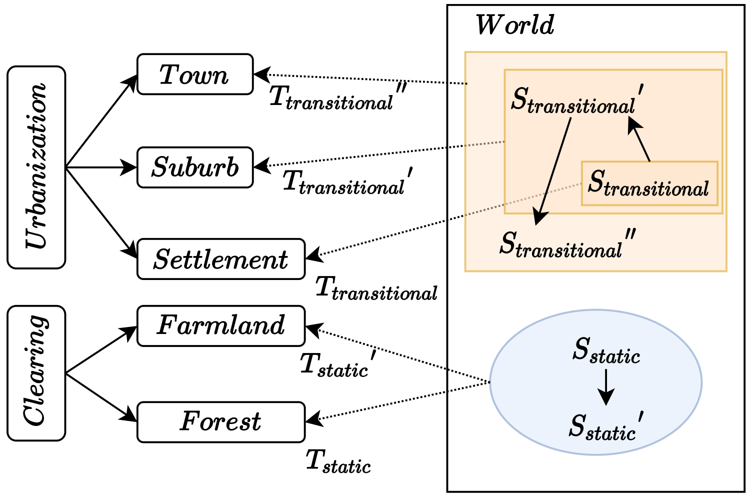

Figure 2: Example spatial change with static *mesasge over space* and transitional *area over space* effects on actions and their change of their spatial characteristics.

data models. The basic concept in conventional GIS data models is the *location*. Basic GIS units are spatial objects such as *points*, *lines*, *polygons* and *cells* and their fixed attributes. Spatial relationships or interactions are limited to being in the same location, in proximity, or to forming relations over sets.

In contrast, natural processes such as specific interactions between individuals and their dynamics, as well as the transformation and translocation of defined system saturates, are modelled. The current GIS extensions for multi-agent simulation is not able to fully support the spatio-temporal modelling of these processes. TGIS should support both location-based analyses, such as the changes of areas or locations as service providers, process-based changes, in the case of the described urbanization or the outbreak of a pandemic (Yergens et al. 2006). The difficulty is that many GIS constructs are developed from the perspective of the structures of spatial data to be provided and do not take into account semantic considerations that can reveal new domain knowledge. For example, the extensive view of a pandemic is more effective by taking into account the cause of the disease, the places of contamination and the spread of the pandemic itself, than only the previously infected areas. That makes it necessary to model these types with their states, events, and processes as an extending construct in a temporal GIS in order to be able to process and query them. Beyond this scope, the semantic analysis of actions within the simulation in terms of their characteristics and behaviour is crucial to determine a set of high-level spatio-temporal constructs to be incorporated into multi-agent simulations. That brings all sub-components to an entirely constructed entity together as exemplary the modelling of a road network for urban simulations, in which a sequence of independent line segments is modelled. These make it difficult to access the whole road itself, instead of processing each sub-road segment.

5 SPATIO-TEMPORAL MODEL

The proposed data model manages both spatial and temporal information and extends the concept of both vector and raster data by a *type-parametric* modelling extension concept on attribute level, allowing data objects within the agent model to access and manage (spatial-)model states over time. For this, we use the *ODMG* (Cattell et al. 2000) type system with its specification of spatial data and the well-known raster model. Time-based information is following the valid dimension according to Bettini et al. (2000) and can be used in multiple *time dimensionality* levels. For multi-agent simulations, moving objects are of great relevance, just as the changes of maps over time described in section 4 are managed.

5.1 Spatio-Temporal Entities

Multi-agent systems use entities to map their respective thematic field. They are based on the principle of entity types, whose instances are bound to a life cycle and are created, modified, and later removed by the simulation system. To illustrate our proposed model in more detail, we describe entities as a class of objects of the same type E_t for entity type with $Et(A_1 : D_1, ..., A_n : D_N)$ where A_i is any attribute and $D_i \subseteq D$ represents the respective value domain of all possible domains D. The domain stores both primitive data types, including *strings*, *integer*, *real*, *booleans* restricted only by the MARS DSL type system, denoted as $\psi \subseteq D$, but can also contain more complex subtypes, encapsulated as independent abstract data types (ADT). Due to this abstraction, we can define both the spatial and temporal domain with their own comprehensive set of operations and predicates (Ong et al. 1983).

The spatial domain $S \subseteq D$ as an abstract data type for an $A_i : S$, which consists of the known objects of lines, polygons, points and cells and which gives an entity its location reference. The temporal domain $T \subseteq D$ with a $A_i : T$ and $T(V_t, T_t, D_t, L_t N_p)$ describes the temporal property of the entity using tri-temporal semantics from temporal databases (Dylla et al. 2013). Here V_t describes the validity time as the point in time or period when a state is valid in the simulation. On the other hand, T_t is the transaction time as the time when a state was actually considered a simulation result, and D_t is the decision time when a decision for action was made. T_t is automatically restricted to never referring to the future. D_t can change in the past, but not in the future.

When linking each version of an entity to preserve the spatial and temporal changes, there is inevitably some additional information to consider. The extending ADT of changes $V(N_p, L_p)$ describes the set of all version changes an entity undergoes. N_p describes the follow-up action, while L_p describes the previous action that created the entity and replaced the previous one. For a concrete $e_t : E_t(A_1 : D_1, ..., A_n : D_n)$ from the domain this is extended for simulation in that $e_t \subseteq D1 \times D2 \times ... \times D_n ... \times T_x \times S_y \times V_z$ applies. Thus, the mapping of any spatial and temporal change concerning V allows forming for each validity period V_t to continue to capture the period when an entity is in a particular version. The entire life span of the entity can also be obtained by the accumulation of subsequent versions. Successive versions of an object can refer to the same spatial properties when changes to the domain take effect, or they can refer to the same subject data when the spatial properties change. This specialization allows asynchronous value changes of the same object within the subject domain or space without mutual influence. A snapshot of a simulation state is still possible using the validity time V_t and the individual lifetime. By using different time granularity, multiple time scales are possible for the same step-based or real-time simulation. Examples of granularity are *seconds*, *minutes*, *days*, *years*. For the step-based simulation scenario, we assume the time domain is a discrete set of times, which are related to the defined time decomposition.

5.2 Spatio-Temporal Operators

From a modelling perspective, we will not propose a whole new modelling interface, but to extend our accepted MARS DSL In order to close the gap of integrating time-related data into multi-agent modelling, we extend our MARS DSL by those *abstract data types* and utilize it within our MARS run-time system. The language provides native agent-based concepts, with restricted type-system and expression language. Types are defined in an object-oriented manner and are encapsulated as an *entity meta type* from the *expressions semantics* processing them. Figure 3 shows an example of the layer and agent definition, extended by the temporal dimension for each affected attribute.

Within the description, both *raster* and *vector layers* are differentiated through an independent meta-type and the explicit geometry (*point, line, region*) is expected for the parameterizable vector layer. Since a *cell* is always used as the underlying geometry for the raster, this definition is not required here. Each attribute describes statically, its temporal granularity level, when a change is expected, with regard to the input configuration of Δt (e.g., $\Delta t = 2$ and $oxide_{value} : real_{days}$ means each two days). The fine-grained distinction is only necessary if a complete layer change is not to take place, as described in section 5.3.

```
raster-layer TemperatureRaster as temperature {
    var temperatureValue : real<hours>
}
vector-layer RiverVector<line> as rivers {
    var waterLevel : real<hours>
}
vector-layer POILayer<point> as points {
    var service : string<days>;
}
vector-layer LanduseVector<region> as landuse {
    var code : integer;
}
```

Figure 3: Example *layer* model definition with temporal change of selected attributes.

To access the time-dimension within the model, we use our existing *simtime* concept and extend it to map each layer- and agent-entity to their historical versions, what then be used by each sub-clause in the model. The time mapping for these entities is defined as a mapping $simtime : E_t \rightarrow \tau \times \{E_{t_\tau}\}$ restricted by the fact that $\forall e \in E_t : e \in E_{t_\tau}$ applies.

As it has been discussed in Worboys (1994), an important requirement for spatio-temporal models is the support for temporal operators. As we use the tri-temporal semantics for entities, as mentioned before in section 5.1, we use Allen's interval logic (Allen 1983) for our model, in order to build joins between temporal data object over the specified relationships. The interval-based temporal logic is defined as follows: Let R be the real line. A temporal interval τ is defined as an ordered pair of time points, with start and endpoint $[start, end] \in R \times R$ and that $start \leq end$. The interval limits are denoted as I_{start} and I_{end}. Allen showed a set of relations applicable to them. These are *overlap*, *precede*, *contain*, *equal*, *meet* and *intersect* operators. Since the interval is a set of ordered and continuous points, sets are the base for interval-oriented reasoning, in which results coming from the *simtime* are always set of versioned entities.

In contrast to the single interval operator, for spatial-object we consider the topological operations specified by OpenGIS, which encompass the *equal*, *disjoint*, *intersect*, *touch*, *cross*, *inside*, *within*, *overlap* and *relate* operators and which are mostly implemented by our MARS DSL. The only metric operators we use is the *distance* with Haversine calculation for geographic data, or euclidean distance when no GIS functionality is used.

5.3 Temporal Layer Modeling

As mentioned in section 3, our approach is based on the principles of GIS layer, in which we integrate the different individual data sets by overlapping them according to their shared geographic coordinate. The definition of a time dimension requires to fix restrictions via the MARS DSL type system. Each associated type of an attribute within an entity type receives a temporal *parameterizable* version in which the time dimension is entered. We define a type as *parameterizable* $\psi < T_d >$ including the desired time dimension $\tau_D = \{seconds, minutes, days, ..., years\}$ with $T_d \in \tau_D$ for which we have the time period to decide when these attributes shall be changed.

In the implementing model, we distinguish between raster 5.3.1 and vector 5.3.2 formats and use a file-based approach with an evolution process to retrieve changes for spatial objects.

5.3.1 Temporal Raster-Layer

Since the standard raster type system (Tennekes 2018) does not support time, we extended this by interrogating time for raster and vector-based data. Raster layers are constructed according to the Space-Time-Composites

(STC) model, already introduced in Ratti and Richens (2004), where the layer for the data source to be addressed, is defined in the concept model. From the outside, the respective source is used in the query. As shown in figure 4, the STC model represents the world as a set of spatially homogeneous and temporally uniform objects in a 2D space. Each space-time composite has its unique temporal progression of attribute changes. Space-time composites can be derived by temporal overlays of time-stamped layers (snapshots) without explicit relations among them (Siabato et al. 2018). The space-time composite conceptually describes the change of a spatial object over time, in which changes are captures in four ways. The *write* directly sets the value at cell (x,y) from the next T_{i+1} input raster. That is done in the following situations: The previous raster cell T_i contains a no data flag; the change is defined explicitly in the metadata description. The *remove* change considers the deletion of value from raster when the cell contains a *NO-DATA-VALUE* flag. And, while the *increment* considers positive update on a raster cell, the *decrement* captures negative ones. Recording of attribute changes is done at discrete simulation time-points, although the temporal resolution is not highly precise. Therefore the STC model can keep temporality within the most generalized units of an attribute, space and time.

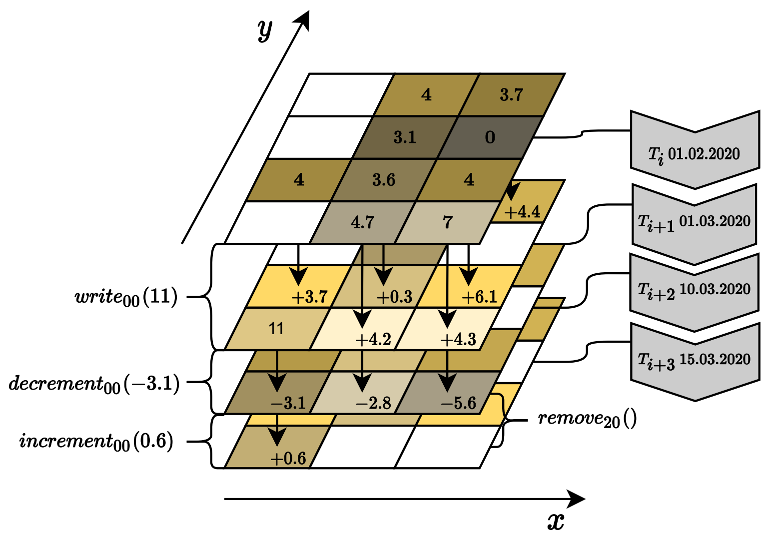

Figure 4: Space-time sparse grid with transition metadata description.

Considering retainment in terms of the number of cells or their extent, MARS applies a transition function $\theta : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R} \times \{true, false\}$. This function localizes changes and their changing type, occurred since the last object version. This incremental approach avoids a re-indexing step of each ST unit. Restructured raster file needs completely to be re-imported, otherwise, including new index creation and read from the source. The result collection for raster files is processed similar to the import. For each cell $c \in R_i$ the δ is collected by the result-adapter, binding them to a complete snapshot, and stored by a sparse-matrix, which delegated to the desired target output (Dalski et al. 2017)

Showing the temporal raster within the model, we use the example urban model defined in figure 3. Within such models, the raster can be used as any other *data-layer*, accessed by their *alias* instance or type. The temporal extension considers the mapping of each data object, using the existing *simtime* concept, described in Glake et al. (2017), where the actual simulation step is inferred previously and applied now onto the spatial object. It returns the raster cell history, in order to answer questions such as “How to find a temperature cell on the raster, which is warm enough in the wintertime to move on?”

Figure 5 shows an example for such an action. The action *moves* the selected agent *entity* by 100 length units, along the target vector, resolved by the query for a cell whose $temperature_{value}$ is within the monthly time range of $[10..2]$. Another option is to build aggregates over the raster. The *explore* concept

```
move entity 100 to nearest on TemperatureRaster where [
    it => it.temperatureValue > 20.0 and
    Time.Month(simtime(it)) => 10
        and Time.Month(simtime(it)) <= 2
]
```

Figure 5: Example time-related *nearest* query for a raster-cell filtered by a condition.

allows us to query multiple cells in a circle, applying the respective aggregate function. Figure 6 shows the aggregation over the raster to get the average temperature within a radius over the last two days.

```
select avg(*) from simtime(
    nearest on TemperatureRaster where [
        cell => cell contains #(xcor,ycor)
    ]
)
```

Figure 6: Example reading and aggregating a raster cell over time by using *explore*.

This query gets each existing value, not marked as *NO-DATA-VALUE*, at the current position of an agent entity (*xcor*, *ycor*) and applies the *avg*(...) function over this time series. For simplicity we could also write *select avg(*) from simtime(temperature.Read(xcor, ycor))* in which we access a limited set of functions provided by the raster alias *temperature*.

5.3.2 Temporal Vector-Layer

In addition to the known raster format, more common spatial structures such as points, lines, and regions (polygons) are available. The temporal *vector-layer* defines the time dimension in two different ways. In addition to the temporal vector model itself, the MARS DSL is extending this processing with multiple query concepts to integrate the new time-dimension into the modelling process. The temporal *vector-layer* is extended by the time dimension similar to the temporal raster in section 5.3.1. Both file exchange and daily index generation are always possible at run-time. In contrast to the linkage of multiple raster maps by a time-series, the time stamp, and multiple versions are managed on attribute level in the associated attribute table. Therefore, the attribute table marks the valid V_t by a respective column.

To illustrate these changes, figure 7 shows the operations currently supported for vector geometry. Besides the change of the whole layer input according to the actual simulation time and the required recreation of the index afterwards, we extend this approach by only change affected data objects. Since the movement of an object is crucial for mobility simulation, the *move* operation takes an input target vector and changes the position within the environment, considering collision checks, e.g., with a river. In contrast, the pos_{at} performs a direct re-positioning of the entity at the desired location. The action prevents intermediates checks of the path towards the target in the form a *line*. The *expand* and *contract* operators use the vector-geometry of *regions* to shrink or grow this site, e.g., the flood is coming. In order to create new objects, the *split* with their reverse function *merge* are provided to change the structure of the regions. In contrast, the *spawn* and *kill* can be used to create and destroy spatial agent entities in the simulation environment.

Movement actions can be defined as shown in figure 8 to solve the problem: *How to move an entity, by distance towards a destination, during a specific-time-period and where an obstacle restricts the path?* The agent is using the *nearest* query, providing a *kNN-query* to get the target position. This example shows the problem of set-based results instead of a single target coordinate. Therefore we infer the most recent data object.

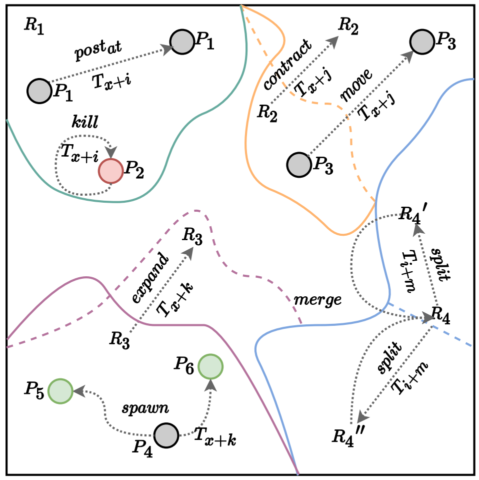

Figure 7: Vector model spatio-temporal transitions for regions (*R*), agents (*A*) and points (*P*).

```
move 10 to nearest on POILayer where
    [it => return it['work'] === "work" ]
        not intersects with River where
            [it => Math.Hour(simtime(it)) <= 7]
```

Figure 8: Example *move* of an agent entity restricted by past obstacles.

We encompass the most well-known types by integrating them into our agent environment and definition set as shown in figure 9. An ordered set of points with distinct start- and endpoint implements the lines. Moreover, each region consists of an ordered set of points linking the start and endpoint. Separate index structures implement points and lines and are used to retrieve the nearest geometries or all within a geometric shape. The so-called Spatial Graph Environment (SGE), introduced in Weyl et al. (2018) and optimized in Weyl et al. (2019), imports line- and point-geometries into a graph structure. The SGE allows agents to move entities along edges concerning obstacles ahead and preventing collisions with other ones. Relational tables, sparse-matrix with Kd-trees for kNN-point query and quad-tree for shape-related queries are used as the data-source and index structure.

```
vector-layer RoadLayer<line> as roads
agent CarDriver on RoadLayer {
    external var identity : integer
    external var profile : DrivingProfile
    external var action : DayplanAction
    observe var velocity : real<seconds>
    observe var acceleration : real<seconds>
    tick { ... }
}
```

Figure 9: Definition of *agent* type with *layer* relationship and *input/output* variables.

In addition to aggregate over raster layer in section 5.3.1, the same operations can be applied on vector as well. As an example, getting the accumulated water level of the river within a recurring time and crossing any agent entity movement, the query showed in figure 10 can be used.

```
(select avg(waterLevel) from explore on RiverVector where [it =>
   Time.Hour(simtime(it)) intersects explore LanduseLayer where [land =>
      not land.restricted and land["type"] == "industry"
   ]
])
```

Figure 10: Vector *sum* aggregation over time for each river in time period *[12h:15h]* and overlapping an *industry* area.

6 CONCLUSION

In this paper, we propose a spatio-temporal data model and their extension in our MARS DSL, to satisfy the problem in closing the gap between temporal and spatial data in multi-agent simulations. Therefore, known problems in the field of temporal GIS were examined, which revealed that not only the snapshot for a single object version is relevant, but also the intermediate actions that led to it. As a result, causal relationships can be resolved, both in terms of domains and spatial structures. These conditions make it necessary to represent processes as well as spatial properties. The presented semantics allows a homogeneous inclusion of spatio-temporal processes and entities into the own modelling perspective. Extensions were integrated as ADT's and be shown how temporal selections and joins are possible on different granularity levels. Based on the importance of querying changes in spatial objects, we have used a temporal versioning mapping, to show how this affects the model.

Further research perspectives include specifying spatio-temporal operators to interact with the database itself and not only with the current and past model state. Spatio-temporal process analysis may also be extended by integrating non-spatial processes related to geographic phenomena (e.g., non-spatial successions or permutations). The evaluation with real-word cases is essential, and besides functionality, also the operator performance has to take into account. A multi-modal simulation to investigate mobility-related problems for the City of Hamburg is, therefore, currently under development.

REFERENCES

Allen, J. F. 1983. “Maintaining Knowledge about Temporal Intervals”. *Communications of the ACM* 26(11):832–843.

Berger, C., M. Bieri, K. Bradshaw, C. Brümmer, T. Clemen, T. Hickler, W. L. Kutsch, U. A. Lenfers, C. Martens, G. F. Midgley, K. Mukwashi, V. Odipo, S. Scheiter, C. Schmullius, J. Baade, J. C. O. du Toit, R. J. Scholes, I. P. J. Smit, N. Stevens, and W. Twine. 2019. “Linking Scales and Disciplines: An Interdisciplinary Cross-Scale Approach to Supporting Climate-Relevant Ecosystem Management”. *Climatic Change* 156(1):139–150.

Bettini, C., S. Jajodia, and S. Wang. 2000. *Time Granularities in Databases, Data Mining, and Temporal Reasoning*. Berlin, Germany: Springer Science & Business Media.

Cattell, R. G. G., D. Barry, M. Berler, J. Eastman, D. Jordan, C. Russell, O. Schadow, T. Stanienda, and F. Velez. 2000. *The Object Data Standard: ODMG 3.0*. San Francisco, California: Morgan Kaufmann.

Dalski, J., C. Hüning, and T. Clemen. 2017. “An Output and 3D Visualization Concept for the MSaaS System MARS”. In *Proceedings of the Agent-Directed Simulation Symposium*, 1–12. Virginia Beach, Virginia: Society for Computer Simulation International.

Dylla, M., I. Miliaraki, and M. Theobald. 2013. “A Temporal-Probabilistic Database Model for Information Extraction”. *Proceedings of the VLDB Endowment* 6(14):1810–1821.

Erwig, M., and M. Schneider. 2002. “STQL—A Spatio-Temporal Query Language”. In *Mining Spatio-Temporal Information Systems*, edited by R. Ladner, K. Shaw, and M. Abdelguerfi, 105–126. Boston, Massachusetts: Springer US.

Glake, D., J. Weyl, C. Dohmen, C. Hüning, and T. Clemen. 2017. "Modeling through Model Transformation with MARS 2.0". In *Proceedings of the Agent-Directed Simulation Symposium*, 1–12. Virginia Beach, Virginia: Society for Computer Simulation International.

Grignard, A., P. Taillandier, B. Gaudou, D. A. Vo, N. Q. Huynh, and A. Drogoul. 2013. "GAMA 1.6: Advancing the Art of Complex Agent-Based Modeling and Simulation". In *International Conference on Principles and Practice of Multi-Agent Systems*, 117–131. Dunedin, New Zealand: Springer.

Jia, P., and A. E. Gaughan. 2016. "Dasymetric Modeling: a Hybrid Approach Using Land Cover and Tax Parcel Data for Mapping Population in Alachua County, Florida". *Applied Geography* 66:100–108.

Lenfers, U. A., J. Weyl, and T. Clemen. 2018. "Firewood Collection in South Africa: Adaptive Behavior in Social-Ecological Models". *Land* 7(3):97.

Mehrotra, R., and A. Sharma. 2009. "Evaluating Spatio-Temporal Representations in Daily Rainfall Sequences from Three Stochastic Multi-Site Weather Generation Approaches". *Advances in Water Resources* 32(6):948–962.

North, M. J., N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. 2013. "Complex Adaptive Systems Modeling with Repast Simphony". *Complex Adaptive Systems Modeling* 1(1):3.

Ong, J., D. Fogg, and M. Stonebraker. 1983. "Implementation of Data Abstraction in the Relational Database System Ingres". *ACM Sigmod Record* 14(1):1–14.

Ratti, C., and P. Richens. 2004. "Raster Analysis of Urban Form". *Environment and Planning B: Planning and Design* 31(2):297–309.

Shekhar, S., Z. Jiang, R. Y. Ali, E. Eftelioglu, X. Tang, V. Gunturi, and X. Zhou. 2015. "Spatiotemporal Data Mining: A Computational Perspective". *ISPRS International Journal of Geo-Information* 4(4):2306–2338.

Siabato, W., C. Claramunt, S. Ilarri, and M. Á. Manso-Callejo. 2018. "A Survey of Modelling Trends in Temporal GIS". *ACM Computing Surveys* 51(2):1–41.

Tennekes, M. 2018. "tmap: Thematic Maps in R". *Journal of Statistical Software* 84(6):1–39.

Weyl, J., D. Glake, and T. Clemen. 2018. "Agent-Based Traffic Simulation at City Scale with MARS". In *Proceedings of the Agent-Directed Simulation Symposium*, 1–9. Baltimore, Maryland: Society for Computer Simulation International.

Weyl, J., U. A. Lenfers, T. Clemen, D. Glake, F. Panse, and N. Ritter. 2019. "Large-Scale Traffic Simulation for Smart City Planning with MARS". In *Proceedings of the 2019 Summer Simulation Conference*, 1–12. Berlin Germany: Society for Computer Simulation International.

Worboys, M. F. 1994. "A Unified Model for Spatial and Temporal Information". *The Computer Journal* 37(1):26–34.

Worboys, M. F. 1998. "Spatio-Bitemporal Geographic Information". In *Spatial and temporal reasoning in geographic information systems*, edited by M. J. Egenhofer and R. G. Golledge, 25–39. Oxford: Oxford University Press.

Xie, X., B. Mei, J. Chen, X. Du, and C. S. Jensen. 2016. "Elite: an Elastic Infrastructure for Big Spatiotemporal Trajectories". *The VLDB Journal* 25(4):473–493.

Yergens, D., J. Hiner, J. Denzinger, and T. Noseworthy. 2006. "Multiagent Simulation System for Rapidly Developing Infectious Disease Models in Developing Countries". In *Proceedings of the 2nd International Workshop on Multi-Agent Systems for Medicine and Computational Biology*, 104–116. Hakodate, Japan: Association for Computing Machinery.

Zhang, C., Q. Yuan, and J. Han. 2017. "Bringing Semantics to Spatiotemporal Data Mining: Challenges, Methods, and Applications". In *2017 IEEE 33rd International Conference on Data Engineering*, 1455–1458. San Diego, California: Institute of Electrical and Electronics Engineers.

AUTHOR BIOGRAPHIES

DANIEL GLAKE is a PhD candidate at the University of Hamburg and has a Master degree in computer science by the University of Applied Science in Hamburg. His research focus areas are polyglot data management, language- and compiler-engineering, including their transformation and more efficient data structures for simulation execution or general-purpose applications. His email address is daniel.glake@uni-hamburg.de.

NORBERT RITTER is a full professor at the Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, University of Hamburg, heading the Databases and Information Systems group. His research focus areas are modern database technology, especially NoSQL-Systems and scalable infrastructures for big data management and cloud data management. His email address is ritter@informatik.uni-hamburg.de.

THOMAS CLEMEN is a full professor in the Department of Computer Science at the Hamburg University of Applied Sciences, Germany. His teaching activities predominantly cover topics within artificial intelligence, information management, and data science. He is the founder and head of the MARS Group (www.mars-group.org), an international and interdisciplinary research group focusing on multi-agent simulation of complex, adaptive, and self-organizing systems. His email address is thomas.clemen@haw-hamburg.de.