Adding Temporal Characteristics to Geographical Schemata and Instances: A General Framework

Morishige Ota[a]

[a] *Kokusai Kogyo Co., Ltd., Tokyo, Japan; morishige_ota@kkc.co.jp*

Abstract: This paper proposes the temporal general feature model (TGFM) as a meta-model for application schemata representing changes of real-world phenomena. It is not very easy to determine history directly from the current application schemata, even if the revision notes are attached to the specification. To solve this problem, the rules for description of the succession between previous and posterior components are added to the general feature model, thus resulting in TGFM. After discussing the concepts associated with the new model, simple examples of application schemata are presented as instances of TGFM. Descriptors for changing properties, the succession of changing properties in moving features, and the succession of features and associations are introduced. The modeling methods proposed in this paper will contribute to the acquisition of consistent and reliable temporal geospatial data.

Keywords: Temporal general feature model, feature instance succession, feature property succession.

1. Introduction

This paper discusses a general framework to add temporal characteristics to geographical schemata and instances. Many approaches to handle the temporal profiles and characteristics of geographical features have been proposed thus far. For example, Snodgrass and Ahn (1986) proposed the use of a heap of snapshots. Langran and Chrisman (1988) proposed a space-time composite comprising polygons with a history attribute distinct from that of their neighbors. Worboys (1994) proposed a unified model for spatio-temporal geometric composites. Mennis, Peuquet, and Qian (2000) proposed the pyramid framework, which consists of both a knowledge component and a data component including temporal attributes. Furthermore, the ISO/TC 211 standardized description rules for 1) temporal characteristics in geographical features (ISO 19108: 2002 – Temporal Schema); 2) moving objects (ISO 19141:2008 – Schema for moving features); and 3) the general feature model (GFM; ISO 19109:2015 – Rules for application schema). The GFM is a meta-model for application schemata. Meanwhile, the Open Geospatial Consortium (OGC) published its implementation standard OCG 14-083r2 – Moving Features Encoding Part1: XML Core in 2015, as an encoding rule for XML documents compliant with ISO 19141.

However, a comprehensive framework comprising 1) an application schema that includes a history of feature types and their relationships, and 2) the succession of feature properties included in an instance, has not yet been proposed. It is not very easy to determine the history directly from the current application schemata, even if the revision notes are attached to the specification. Therefore, we will first propose a framework to add temporal characteristics to application schemata. Then, we will propose an approach to describe successions of spatial and non-spatial properties in a feature instance. Finally,

the functional enhancement of the geographic information technology tool kit named gittok (Ota and Plews, 2015) will be discussed, as an example of a software implementation of the proposed framework.

2. Application schemata with time-dependent components

The GFM defines four types of components: the feature, association, attribute, and operation types. Given that most application schema components should have a life cycle, all four component types should be defined as time-dependent components in the schema, and should therefore possess temporal attributes. For example, a validity period on feature and attribute types indicates that they can be used for instantiations only during the indicated period of time. Similarly, a validity period on an operation type means that this operation type can only be used to obtain a return value during the indicated period of time. An enhanced GFM with the capability of representing the temporal characteristics of the components will be proposed and called a temporal GFM (TGFM) in this paper. It will be discussed in detail in Section 4.

Versioning has been commonly used to revise specifications; usually, a new specification will include the revision notes. However, it is not easy to determine exactly which parts were revised directly from the application schema, because the components carry no information about revisions. To solve this problem, rules for the description of the succession between different components are required. An application schema capable of describing such a succession will enable the retrieval of the past or future versions of any component, and the valid application schema at any point of time can be extracted as a time slice of the base application schema. To achieve such a capability, each component should have a validity period, and the temporal relationships between different components should be represented as

topological relationships between periods. These requirements can be implemented by referring to the temporal schema defined in ISO 19108. According to this standard, each period has beginning and ending instants shared with its immediate neighbors. Validity periods are represented by the TM_Period primitive, and the succession is represented by the "Beginning" and "Ending" associations shown in "Figure 3 - Temporal geometric primitives" in ISO 19108.

3. Succession of properties and associations

According to the temporal schema provided by ISO 19108, temporal primitives form a directed acyclic graph—in other words, a chain of changes. There are two levels of change. The first level consists of changes in the property values. They can be both quantitative (e.g., position, shape, and temperature) and qualitative (e.g., owner's name and person's address). We call them "changing properties." A changing property may be a temporal sequence of more than one time slice element; it may also refer to one single element and its validity period. It applies to properties that are "static" during a certain period of time (e.g., owner's name and person's address); being "static" can, however, be thought of as a special case of change. The second level is a succession of these changing properties. It forms a directed acyclic graph in which edges are equivalent to the changing properties. For example, let's think about the movement of a person who walked from home, took a tram, and afterward took an airplane to travel abroad. We can describe the trajectory of this person as one curve. However, considering that the characteristics of the used transportation means are very diversified, the curve should be separated in sections, to reflect the differences in transportation modes. Such a temporal succession of changing properties describes the notion of moving.

Meanwhile, operations may also evolve in time. It happens, for example, when a better algorithm is found, or when a different algorithm is chosen as a response to new social demands. Similarly, associations may also evolve in time. It happens, for example, when a person buys a second house: the association between this person and the set of houses changes from one-to-one to one-to-two.

To describe these phenomena, we should first define changing properties possessing a validity period. Then, we should define a succession of changing properties to describe the temporal evolution of a feature and the temporal association between features.

4. Software implementation framework

We should consider a temporal extension of GFM and find a way to describe successions of changing properties and of implementing them in geographic information software containing the capability for describing application schemata and feature instances. However, the GFM provided by ISO 19109 does not specify a way to describe the life cycle of an application schema. Additionally, the moving feature model provided by ISO 19141 and OGC 14-083r2 does not specify

successions of changing properties. We will discuss both issues in this section.

4.1 Temporal general feature model (TGFM)

If every element of an application schema has an assigned time validity, a time slice of the application schema can be constructed. Fig. 1 illustrates the TGFM used in the geospatial information technology tool kit named gittok (Ota and Plews, 2015). The TGFM includes an abstract class called PrimitiveType, to be able to describe the validity period of types such as FeatureType, AssociationType, AttributeType, and OperationType. This is made possible by using the optional attribute "validTerm." If this attribute is not declared, the type is valid as long as the application schema is valid.
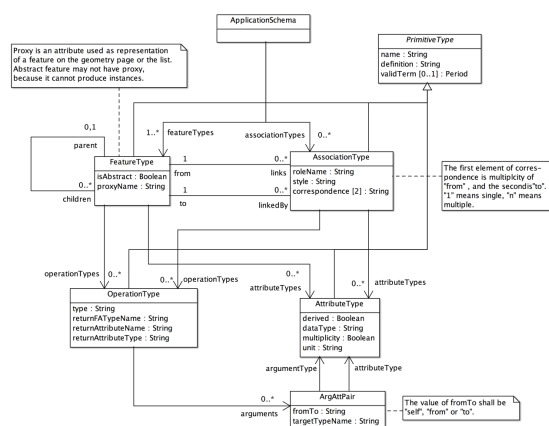

Fig. 1. UML diagram describing the temporal general feature model (TGFM).

However, the succession between a type and its succeeding type is not made clear by the TGFM, because time is a total order set. In general, every period has succession relationships with the contiguous periods. Therefore, the succession between different types—such as a feature type and an operation type—cannot be prohibited only by the TGFM. To avoid the inconsistency, we should describe the relations between periods selectively, based on rationality. This is possible if we assume that the topological time is partially ordered. In other words, by allowing the definition of more than one temporal edge or node at the same geometric interval or instant. In this context, the geometric dimension of time is one, but the topological dimension is more than one. As a consequence, topological time can be described by a directed acyclic graph. The temporal schema provided in ISO 19108 supports this hypothesis (Figure 6 - Topology of time in ISO 19108:2002). Figs. 2, 3, 7, and 9 in this paper are examples of temporal application schemata.

The imaginary example shown below describes the history of the feature type defining "Forestland." This feature type has been valid since 1956-10-23, when the oldest forestland record was stored in the master file; the digital file was made later, of course. The attribute "legal regulation" was added to the feature type definition in 1981-05-24. The rule for the description of "legal regulation" was changed from a free character string to an

identifier defined in the forest conservation code in 1998-09-12. In the schema presented below, the succession from "att002" to "att003" is clarified by the connection between the edges "p001" and "p002" provided by the node "i002." In accordance with this schema, in 1990 the legal regulation was described by the free character string, because the valid attribute type was "att002" (the validity period is from 1981 to 1998).

```
// Feature type definition
FeatureType =
        {"name": "Forestland", "definition": "an
        area of land covered by forests",
        "validTerm": "p000", "isAbstract": "false",
        "proxyName": "extent", "attributeTypes":
        ["att000", "att001", "att002", "att003"]}


// Attribute type definitions AttributeType["att000"] =
        {"name": "owner", "definition": "a person
        who owns the forest land.", "validTerm":
        "p000", "derived": false, "dataType":
        "String", "multiplicity": false, "unit": "",}


AttributeType["att001"] =
        {"name": "extent", "definition": "a
        geometry covered by something.",
        "validTerm": "p000", "derived": false,
        "dataType": "SG_Surface", "multiplicity":
        false, "unit": "",}


AttributeType["att002"] =
        {"name": "legal regulation", "definition":
        "a rule made by an authority for the forest
        conservation.", "validTerm": "p001",
        "derived": false, "dataType": 'String",
        "multiplicity": true, "unit": "",}


AttributeType["att003"] =
        {"name": "legal regulation", "definition":
        "a rule made by an authority for the forest
        conservation.", "validTerm": "p002",
        "derived": false, "dataType": "Forest
        conservation code list",
        "multiplicity": true, "unit": "",}


//Temporal topology
Period["p000"] = {"begin": "i000", "end": "i999"}
Period["p001"] = {"begin": "i001", "end": "i002"}
Period["p002"] = {"begin": "i002", "end": "i999"}


//Geometric realizations of nodes
Instant["i000"] = {"date": "19561023", "tense": "past"}
Instant["i001"] = {"date": "19810524", "tense": "past"}
Instant["i002"] = {"date": "19980912", "tense": "past"}
Instant["i999"] = {"date": "", "tense": "present"}
```

4.2 Succession of properties, associations, and instances

The topological time can be represented as a directed acyclic graph; the correspondences between previous and posterior periods are: substitution (1:1), division (1:n), and fusion (m:1) (ISO 19108:2002). Generation (0:1) and extinction (1:0) are also considered to be special cases of succession. All m:n relationships and separations—such as child bearing—can be obtained as a composition of the primitive succession types listed above. A set of contiguous successions forms the directed acyclic graph to describe the temporal topology. Four examples will be presented below: succession of a static property, succession of a moving property, succession of an association, and succession of instances.

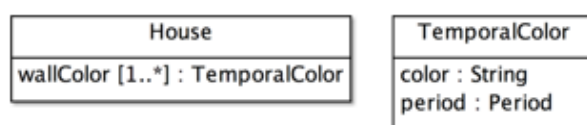Example 1. A house was repainted from gray to white.

Fig. 3. Application schema describing the wall color change.

Fig. 4. Illustration describing the wall color change.

```
//Attribute value
wallColor = [{"color": "gray", "period": "p001"},
{"color": "white", "period": "p002"}]


//Temporal topology
period["p001"] = {"begin": "i001", "end": "i002"}
period["p002"] = {"begin": "i002", "end": "i999"}


//Geometric realizations of nodes
instant["i001"] = {"date": "19900725", "tense": "past"}
instant["i002"] = {"date": "20150312", "tense": "past"}
instant["i999"] = {"date": "", "tense": "present"}
```

Example 2. In the morning of December 9, 2017, a primary school child walked to the bus stop, waited for a while, and then took a school bus to go to school.
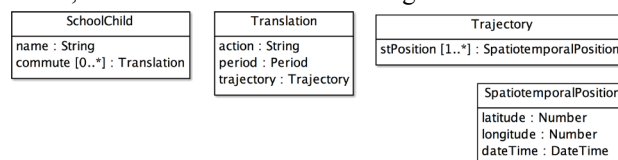

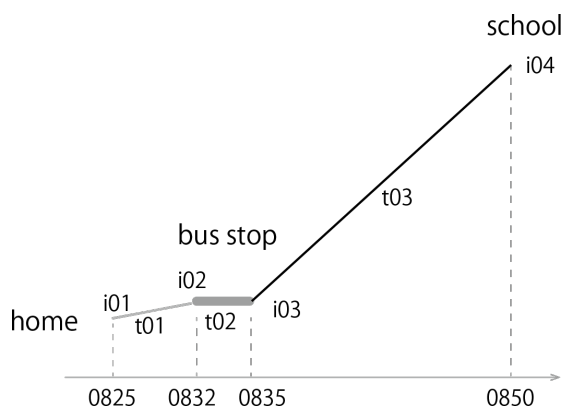Fig. 5. Application schema describing the commute of a primary school child.

Fig. 6. Illustration describing the spatio-temporal evolution of the commute.

```
//Attribute value
commute = [{"action": "walking", "period": "p01",
"trajectory": "t01"},
{"action": "waiting", "period": "p02", "trajectory":
"t02"},
{"action": "taking a school bus", "period": "p03",
"trajectory": "t03"}]
```

```
//Temporal topology
period["p01"] = {"begin": "i01", "end": "i02"}
period["p02"] = {"begin": "i02", "end": "i03"}
period["p03"] = {"begin": "i03", "end": "i04"}
```

```
//Geometric realizations of nodes
instant["i01"] = {"dateTime": "20171209T0825", "tense":
"past"} instant["i02"] = {"dateTime": "20171209T0832",
"tense": "past"} instant["i03"] = {"dateTime":
"20171209T0835", "tense": "past"} instant["i04"] =
{"dateTime": "20171209T0850", "tense": "past"}
```

```
//Trajectory trajectory["t01"] =
[{"latitude": 35.09896, "longitude": 139.08926,
"dateTime": "20171209T0825"},
.......
{"latitude": 35.09928, "longitude": 139.08934,
"dateTime": "20171209T0832"}]
```

```
trajectory["t02"] =
[{"latitude": 35.09928, "longitude": 139.08934,
"dateTime": "20171209T0832"},
{"latitude": 35.09928, "longitude": 139.08934,
"dateTime": "20171209T0835"}]
```

```
trajectory["t03"] =
[{"latitude": 35.09928, "longitude": 139.08934,
"dateTime": "20171209T0835"},
.......
{"latitude": 35.11056, "longitude": 139.116521,
"dateTime": "20171209T0850"}]
```

Example 3. Akiko Sato, who lives in an apartment in Tokyo, purchased a cottage in Karuizawa. Therefore, she has two real estate properties at present.
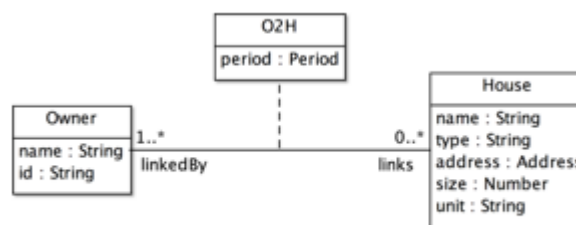

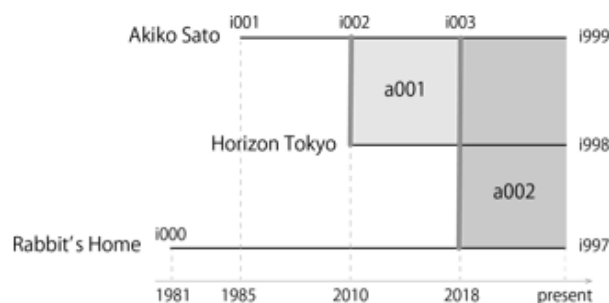Fig. 7. Application schema describing an association between owners and houses.

Fig. 8. Illustration describing the associations between Akiko Sato and her two houses.

```
// Owner instance
owner = {"name": "Akiko Sato", "id": "6548", "links":
["a001", "a002"], "period": "p000"}
```

```
// House instances house["h001"] =
{"name": "Horizon Tokyo", "type": "apartment",
"address": "109-1008 Shinjuku-Ku, Tokyo,162-0835",
"size": 102, "unit": "square meter", "linkedBy": "a001",
"period": "p100"}
house["h002"] =
{"name": "Rabbit's Home", "type": "cottage", "address":
"18-2-30 Karuizawa, Nagano, 389-0802", "size": 55,
"unit": "square meter", "linkedBy": "a002", "period":
"p101"}
```

```
// Association between person and houses
o2h["a001"] = {"linkedBy": "6548", "links": ["h001"],
"period": "p200"}
o2h["a002"] = {"linkedBy": "6548", "links": ["h001",
"h002"], "period": "p201"}
```

```
// Temporal topology
period["p000"] = {"begin": "i001", "end": "i999"}
period["p100"] = {"begin": "i002", "end": "i998"}
period["p101"] = {"begin": "i000", "end": "i997"}
period["p200"] = {"begin": "i002", "end": "i003"}
```

period["p201"] = {"begin": "i003", "end": "i998"}

```
// Geometric realizations of nodes
instant["i000"] = {"date": "19810725", "tense": "past"}
instant["i001"] = {"date": "19850312", "tense": "past"}
instant["i002"] = {"date": "2010701", "tense": "past"}
instant["i003"] = {"date": "20180312", "tense": "past"}
instant["i999"] = {"date": "", "tense": "present"}
instant["i998"] = {"date": "", "tense": "present"}
instant["i997"] = {"date": "", "tense": "present"}
```

Example 4. One parcel was divided into two, as a response to a partial item sale.

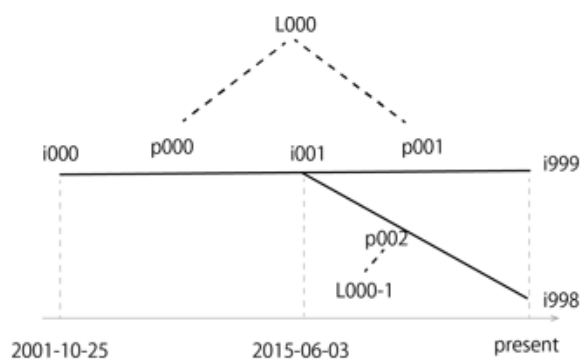Fig. 9. Application schema describing the parcel history.

Fig. 10. Illustration describing the parcel division.

```
// Parcel instances
parcel["L000"] = [{"owner": "Jun Fuyuki", "size": 253.3,
"unit": "square meter", "period": "p000"},
{"owner": "Jun Fuyuki", "size": 132.5, "unit": "square
meter", "period": "p001"}]
parcel["L000-1"] = [{"owner": "Tomoya Kato", "size":
120.8, "unit": "square meter", "period": "p002"}]
```

```
// Temporal topology
period["p000"] = {"begin": "i000", "end": "i001"}
period["p001"] = {"begin": "i001", "end": "i999"}
period["p002"] = {"begin": "i001", "end": "i998"}
```

```
// Geometric realizations of nodes
instant["i000"] = {"date": "20011025", "tense": "past"}
instant["i001"] = {"date": "20150603", "tense": "past"}
instant["i999"] = {"date": "", "tense": "present"}
instant["i998"] = {"date": "", "tense": "present"}
```

5. Conclusion and future work

This paper proposed the temporal general feature model (TGFM) as a meta-model for application schemata, which enables modeling the changes and time evolution of real-world phenomena. Simple examples of application schemata were presented, as instances of TGFM. Furthermore, sample descriptions for changing properties, the succession of changing properties in moving features, and the succession of feature associations and instances were introduced. The modeling methods proposed in this paper will contribute to the acquisition of consistent and reliable time-referenced geospatial data.

However, without actual software implementation and experimental validation, it is impossible to prove the validity of the proposals. The capability to describe the time-related characteristics of real world phenomena will be implemented on a geospatial information technology tool kit named gittok (Ota and Plews, 2015). Gittok has been developed based on geographic information standards, and it has been used to teach Geographic Information Technology in both higher education and continuing education programs in Japan. Gittok can also be seen as a platform for evaluating new concepts such as the temporal modeling proposed in this paper. However, the validity of the ideas proposed in this paper cannot be proved without a capability to show the dynamic illustrations corresponding to the temporal features and associations. Therefore, not only the modeling method, but also the representation method must be implemented in gittok. This will be the object of future work.

6. Acknowledgements

7. References

ISO, ISO 19108 (2002). Geographic information – Temporal schema. ISO.

ISO, ISO 19109 (2015). Geographic information – Rules for application schema. ISO. ISO, ISO 19141 (2008). Geographic information – Schema for moving features. ISO.

Langran, G., and Chrisman, N. R. 1988. A framework for temporal geographic information. Cartographica: The International Journal for Geographic Information and Geovisualization, 25(3), 1–14.

Mennis, J. L., Peuquet, D. J., and Qian, L. 2000. A conceptual framework for incorporating cognitive principles into geographical database representation.

International Journal of Geographical Information Science, 14(6), 501–520.

OGC, 14-083r2 (2015) Moving Features Encoding Part1: XML Core, OGC.

Ota, M., and Plews, R. 2015. Development of a software tool as an introduction to Geospatial Information Technology based on geospatial standards. Cartography and Geographic Information Science, 42(5), 419–434.

Snodgrass, R. T., and Ahn, I. 1986. Temporal databases. Computer, 19(9), 35–42.

Worboys, M. F. 1992. Object-oriented models of spatiotemporal information. In ASPRS GIS/LIS International Conference. Vol. 2, pp. 825–834.