TGML - Extending GML by Temporal Constructs - A Proposal for a Spatiotemporal Framework in XML

Alexander Zipf
European Media Laboratory -EML,
Heidelberg, Germany
++49 - 6221- 533 202
alexander.zipf@eml.villa-bosch.de

Sven Krüger
European Media Laboratory -EML,
Heidelberg, Germany
++49 - 6221 - 533 227
sven.krueger@eml.villa-bosch.de

ABSTRACT

XML is nowadays a commonly used way for describing and exchanging data. Within the GIS community the OpenGIS Consortium (OGC) recently has published the Geographic Markup Language (GML) as a new specification [8]. GML is essentially an XML encoding of the Simple Feature Specification (SFS). A range of further XML-based standardization efforts by the OGC are under development right now. But it is known that spatial features do not only have geometric or thematic properties, but also temporal aspects. These have been neglected by the OGC and most GIS vendors so far, but have been an active area of research for years. Recently a flexible object-oriented temporal framework for describing 4D-geoobjects has been developed [21]. This temporal framework has been realized in Java and has been implemented on two databases (namely the OODB Jasmine by Computer Associates and the OR DB Cloudscape by Informix). The temporal framework is a self-consistent object-oriented structure to describe temporal data and can easily be represented by an XML schema. This schema can be combined with the existing GML schema to realize a powerful spatio-temporal XML-Schema. The temporal XML schema is being introduced and explained in this paper, and examples using a geo-server are being presented. The latter has been developed for the Deep Map project and implements the SFS for CORBA interfaces [19], as well as it does also support GML.

Categories and Subject Descriptors

H. Information Systems
H.2 DATABASE MANAGEMENT
H.2.1 Logical Design

General Terms

Standardization

Keywords

Geographic Markup Language (GML), GIS, spatio-temporal models, databases, XML-schema

ACM-GIS 2001, November 9-10, 2001, Atlanta, GA, USA.

1 INTRODUCTION

Within the Deep Map project by the European Media Laboratory (EML) [7],[20] a flexible and extensive temporal object-oriented model had been developed. The original aim was to allow the management of 3D geo-objects of urban areas over historic epochs and acts as basis for the data management components of temporal 3D-GIS to be developed in the future ("4D-GIS" or "3D-TGIS"). Thus it contributes to one of the long term goals of the Deep Map project – namely "virtual time travels". But since the temporal part of this model is a self-consistent OO-model for temporal structures it can also be used with purely 2D-geodata as described here. In this contribution we explain how the complex aspects of time can be modeled and described using XML-schema and how to combine this with GML. For the resulting schema we propose the name TGML - "Temporal Geographical Markup Language". The technical aspects of the realized prototype include a 3-tier architecture using java that guarantees a high degree of flexibility and platform independence.

2 ASPECTS OF SPATIO-TEMPORAL DATA MODELLING

A geo-object or feature in general consists of the aspects theme, geometry, topology and time [1][14]. Still today's GIS don't handle all aspects equally well. The temporal dimension is definitely an important aspect of almost all phenomena of the real world. Traditional databases as well as GIS delivered only a snapshot of the real, therefore there was a need for new data models that allows the handling of temporal data [13],[15]. Within the last years a range of temporal models were also developed in the field of object-oriented databases. [9][10] and [11] give a summary. E.g. [3],[6],[16] and [18] present possibilities for an object-oriented integration of temporal models into 2D GIS. To represent the basic elements of the temporal framework some important concepts are defined briefly. The period of the physical process used to measure time is called "chronon" while the duration of the period is described as a "granularity. A flexible temporal framework should provide means to representing quite arbitrary calendars and their granularities and some further aspects of time as explained below in more detail [5].

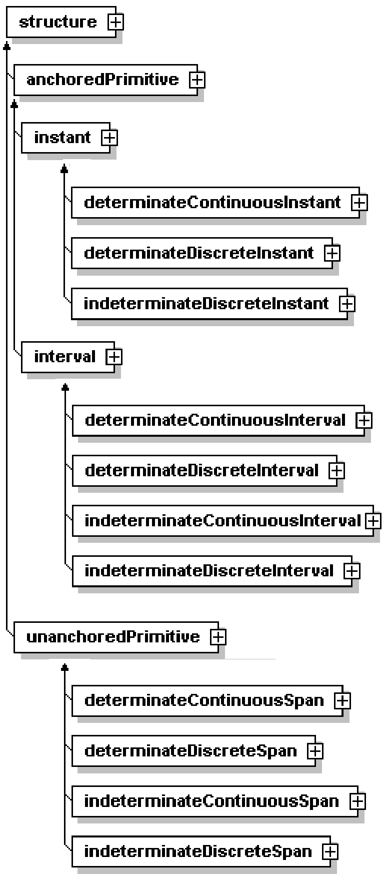
3 AN OBJECT-ORIENTED MODEL FOR TEMPORAL DATA

The object-orientated paradigm is used here for the modeling of a general time framework. The range of possible different applications put quite complex requirements on the temporal support. The dimensions that limit the modeling space of a general temporal model have to be identified as well as their components and properties have to be determined in order to allow the definition of an adaptable structure that can fulfill these various nition of an adaptable structure that can fulfill these various requirements. Using these identified components, a framework for building temporal models was developed that supports design alternatives by provision of classes and accompanying properties. The following general aspects can be distinguished:

- *Temporal Structure* – defines a structure using temporal primitives, domains and structures concerning temporal determination (certain or uncertain representations).
- *Temporal Order* - describes the possible types of orders of temporal structures.
- *Temporal History* - describes the semantic meaning of the different states the object.
- *Temporal Representation* - describes how to represent calendars and granularities.

3.1 Temporal Structure

The temporal structure defines the building blocks of a temporal model. The figure shows the hierarchical structure of the individual components of a temporal structure.


structure
anchoredPrimitive
instant
determinateContinuousInstant
determinateDiscreteInstant
indeterminateDiscreteInstant
interval
determinateContinuousInterval
determinateDiscreteInterval
indeterminateContinuousInterval
indeterminateDiscreteInterval
unanchoredPrimitive
determinateContinuousSpan
determinateDiscreteSpan
indeterminateContinuousSpan
indeterminateDiscreteSpan

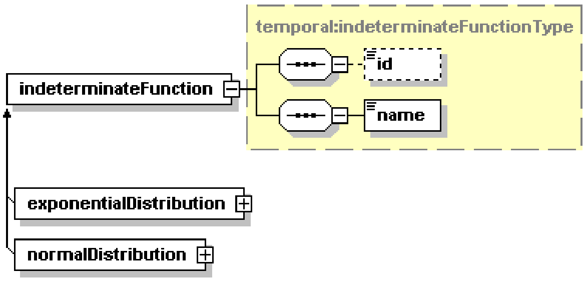
```
<indeterminateDiscreteInterval>
    <id>1000001</id>
    <logicalRepresentation>1945-1946</logicalRepresentation>
    <gregorianCalendar>
        <id>1000001</id>
    </gregorianCalendar>
    <belongsTo>
        <linearOrder>
            <id>1000001</id>
        </linearOrder>
    </belongsTo>
    <beginChronons>14041123200</beginChronons>
    <durationBegin>31536000</durationBegin>
    <endChronons>14072659200</endChronons>
    <durationEnd>31536000</durationEnd>
    <normalDistribution>
        <id>1000001</id>
    </normalDistribution>
</indeterminateDiscreteInterval>
```

Fig. 1: Design choices of a temporal structure and example

This temporal "structure" can have the following properties [12]:

1. *Temporal Primitives* are represented either as absolutes (anchored, "date", e.g.: 5-9-1999) or relative (unanchored, "period of time", e.g.: 30 days).
2. *Temporal domain:* It is possible to distinguish discrete and continuous domains. In the field of temporal databases a discrete time domain is usually being used.
3. *Temporal determination*: In the deterministic case complete and exact knowledge is available for temporal primitives. On the other hand these aren't determined exactly in indeterministic cases [4], e.g. fuzzy temporal borders.

In the last example the provision of the parameters defining the fuzzy membership function would be needed as additional tags. In figure 2 two possibities are displayed.

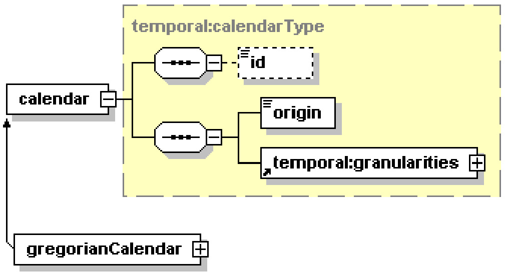


```
<normalDistribution>
    <id>1000001</id>
    <name>standard normal distribution</name>
    <parameters>
        <parameter name="μ">0</parameter>
        <parameter name="sigma">1</parameter>
    </parameters>
</normalDistribution>
```

Fig. 2: indeterminate function model and example

The topmost level of the temporal structure-model consists of absolute (anchored) and relative (unanchored) temporal primitives. The next hierarchical level supplements the structure with domains, being either discrete or continuous. The deterministic and not-deterministic primitives form the last component. A temporal structure consists of a combination of all of the represented temporal primitives. By combining the components of the three levels there are eleven temporal primitives possible in order to model temporal phenomena.

3.2 Temporal Representation



```
<gregorianCalendar>
    <id>1000001</id>
    <origin>1582</origin>
    <granularities>
        <granularity>
            <name>year</name>
            <minimumValue>0</minimumValue>
            <maximumValue>99</maximumValue>
        </granularity>
    </granularities>
</gregorianCalendar>
```

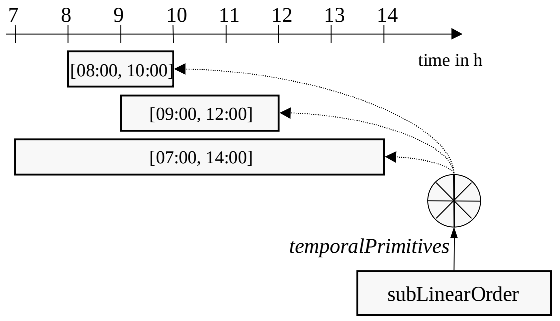
Fig. 3: temporal representation model and example

The temporal primitives represent the basis for a representation of temporal data. It is necessary to distinguish between the logical and physical representation of a time value. If the time value is described by means of a calendar, it is a logical representation.

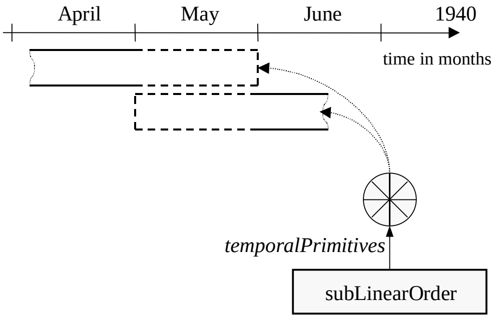
3.3 Temporal Order

The course of the time can be classified as linear, sublinear or branching. In the linear case overlapping borders of temporal primitives are forbidden, while possible in the sub-linear case. A sub-linear order can also be used for managing indeterministic temporal phenomena, e.g. for describing the changes of an object which are only known indeterminate.

Time is regarded to be linear only to a certain point of time within a branching order. A typical example would be town planning, where different planning alternatives can be managed in different branches of the resulting temporal tree.



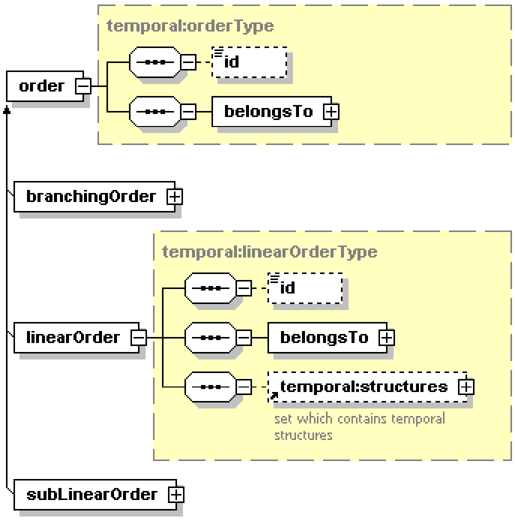
-fig A-



-fig B-

Fig. 4: Example for the use of a sub-linear order

Example: A fire destroyed the roof of a house sometime in May 1940. To model the two different conditions (house before/after fire within/without roof) two non-deterministic intervals can be used due to temporal uncertainty. These have overlapping borders, since it isn't exactly clear when the change of condition – in this case the fire - did happen (fig. 4-B).

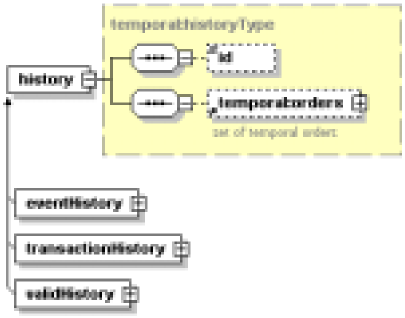


```
<linearOrder>
    <id>1000001</id>
    <belongsTo>
        <validHistory>
            <id>1000001</id>
        </validHistory>
    </belongsTo>
    <structures>
        <determinateDiscreteInstant>
            <id>1000001</id>
        </determinateDiscreteInstant>
        <determinateDiscreteInterval>
            <id>1000002</id>
        </determinateDiscreteInterval>
        <indeterminateDiscreteInterval>
            <id>1000003</id>
        </indeterminateDiscreteInterval>
    </structures>
</linearOrder>
```

Fig. 5: temporal order model and example

3.4 Temporal History Type

One of the requirements for a temporal model is to represent the development of real world objects over time. This development forms the "temporal history" of the object and can be distinguished between in *valid time* and *transaction time* [13].



```
<validHistory>
  <id>1000001</id>
  <orders>
     <linearOrder>
        <id>1000001</id>
     </linearOrder>
     <linearOrder>
        <id>1000002</id>
     </linearOrder>
     <subLinearOrder>
        <id>1000003</id>
     </subLinearOrder>
  </orders>
</validHistory>
```

Fig. 6: temporal history model and example

3.5 Putting the Components Together

The necessary elements and design alternatives for an object-oriented temporal model have been explained so far. The following concentrates on the relations between these components.

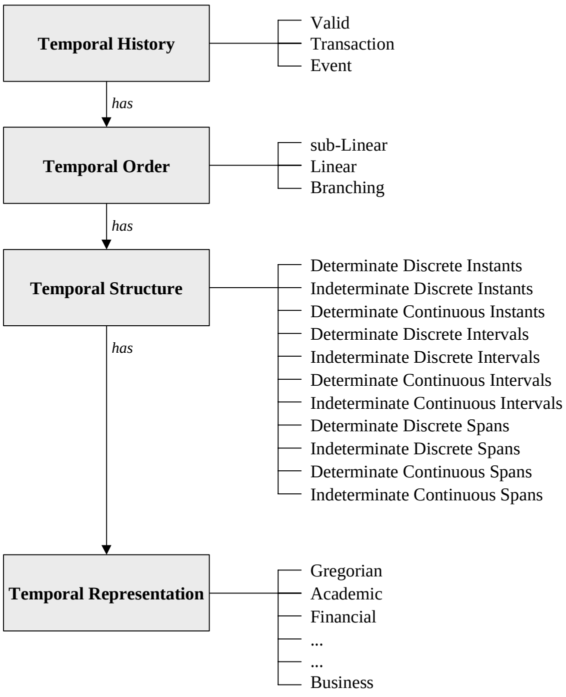

Temporal History
Valid
Transaction
Event
has
Temporal Order
sub-Linear
Linear
Branching
has
Temporal Structure
Determinate Discrete Instants
Indeterminate Discrete Instants
Determinate Continuous Instants
Determinate Discrete Intervals
Indeterminate Discrete Intervals
Determinate Continuous Intervals
Indeterminate Continuous Intervals
Determinate Discrete Spans
Indeterminate Discrete Spans
Determinate Continuous Spans
Indeterminate Continuous Spans
has
Temporal Representation
Gregorian
Academic
Financial
...
...
Business

Fig. 7: Primitives for a temporal model

A temporal model can support one or several history types (like “valid time” or “transaction time”). Each of these history types consists of a set of temporal orders (with either linear, sub-linear or branching properties). In contrast to the linear order it is possible for sub-linear or branching orders that their (absolute) temporal primitives overlap and form several partial temporal orders. Each of these temporal orders contains a temporal structure, that consists of either all or a subset of the eleven different temporal primitive presented in illustration 1. To offer highest flexibility, different calendars can be defined and assigned to the temporal primitives.

4 INTEGRATING GML WITH THE TEMPORAL FRAMEWORK

The object oriented-paradigm allows to model the different aspects of features and the different temporal constructs as own classes that can be related to each other.

By the assignment of a temporal order to every time variant feature or time variant attribute it is possible to model temporally changing spatial data. Object-time-stamping can be used to describe the changes of an complex object over time. In this case the whole object (e.g. gml-feature) will be time-stamped by adding a reference to an order. Figure 8-A illustrates how such a link could be realized.

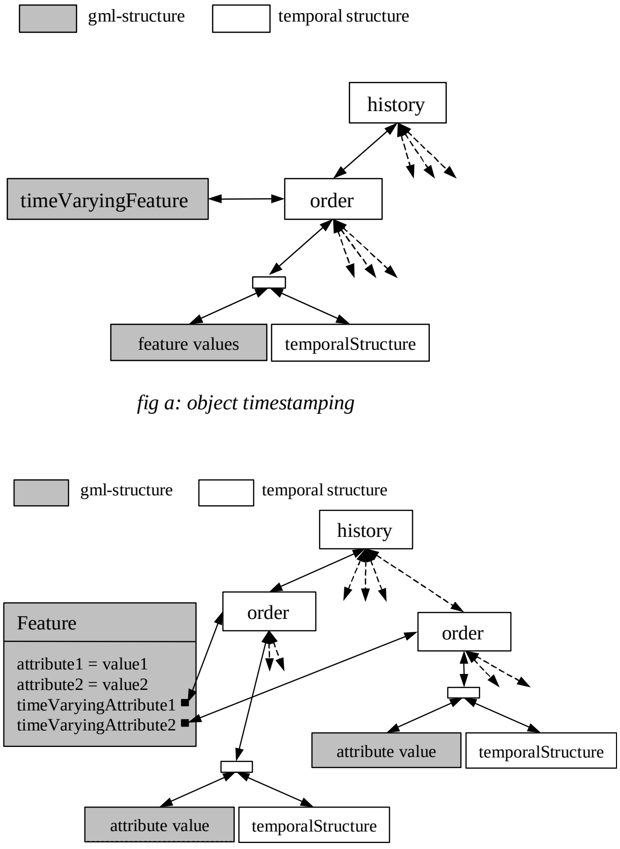


fig a: object timestamping

fig b: attribute timestamping

Fig. 8: coupling gml- and temporal-structures

If an object has only a few time-invariant-attributes attribute-time-stamping may be more efficient regarding to reduce redundant data structures. In this case every time variable attribute is extended by a reference to an order (fig. 8-b). More detailed examples can be found in the appendix. By adding a reference to a linear order the first example describes how the time varying values of the gml-object '<_Feature>' can be modeled (object-time-stamping). The second one shows the realization if only the '_Geometry' - attribute of the gml-structure '<_Feature>' is time variant. This can be applied to any attribute of the gml-structure, so that every attribute can have its own temporal constructs.

5 CONCLUSIONS

For an information system like Deep Map covering the aspects of town history and also for a multitude of other applications, the management of changes in feature data (e.g. buildings) is of particular importance. But since the temporal model introduced here is very generic, in spite of being developed for the Deep Map "4D" GIS in the first place it can also be exploited in completely different applications that lack of spatial properties. Within this paper we explained that this framework can cover both valid-time histories as well as transaction-time histories of features described using OGC's GML in a very flexible way, as it provides the necessary building blocks for temporal structures like *intervals*, time *spans* or *instants*. Each of these can be subdivided being either *continuous* or *discrete* on the one hand and *determinate* or *indeterminate* on the other hand. By supporting the definition of additional application specific calendars with their respective granularities the framework supports all the notions and notations of time we consider relevant for present practical applications.

6 ACKNOWLEDGMENTS

This work has been undertaken in the context of the project Deep Map at the EML, supported by Klaus Tschira Foundation (KTS) and the EU funded project CRUMPET (IST-1999-20147).

7 REFERENCES

[1] BILL, R. und FRITSCH, D. (1991): Grundlagen der Geoinformationssysteme. Bd.1, Hardware, Software und Daten. Wichmann. Heidelberg.

[2] BØHLEN M., JENSEN C. and SKJELLAUG B. (1998): Spatio-Temporal Database Support for Legacy Applications, In: Proceedings of the 1998 ACM - Symposium on Applied Computing, February 27-March 1, 1998. Atlanta, GA. 226-234.

[3] BREUNIG, M. (2000): On the way to component-based 3D/4D geoinformation systems. In: Lecture Notes in earth Sciences. Vol 94. Springer. Heidelberg.

[4] DYRESON, C.E. und SNODGRASS, R. (1993): Valid-time Indeterninacy. Proceedings of the 9th International Conference on Data Engineering. 335-343.

[5] KRÜGER, S. (2000): Konzeption und Implementierung eines temporalen objekt-orientierten Modells für 3-dimensionale Geo-Objekte im Deep Map-Projekt. Diplomarbeit. Institut für Praktische Informatik und Medieninformatik, Technische Universität Illmenau.

[6] LANGRAN, G. (1989): Time in Geographic Information Systems. Taylor & Francis Ltd. 1992.

[7] MALAKA, R. and ZIPF, A. (2000): Deep Map - Challenging IT research in the framework of a tourist information system. In: Fesenmaier, D. Klein, S. and Buhalis, D. (Eds.): Information and Communication Technologies in Tourism 2000. (Proceedings of ENTER 2000, 7th. International Congress on Tourism and Communications Technologies in Tourism. Barcelona. Spain). Springer Computer Science, Wien, New York. pp 15-27.

[8] OPEN GIS CONSORTIUM (2001): The Geographic Markup Language (GML) Recommendation 2.0. http://www.opengis.org/techno/specs.htm

[9] SKJELLAUG, B. (1997a): Temporal Data: Time and Relational Databases. Research Report 246. Department of Informatics, University of Oslo.

[10] SKJELLAUG, B. (1997b): Temporal Data: Time and Object Databases. Research Report 245. April 1997. Department of Informatics, University of Oslo.

[11] SKJELLAUG, B. and BERRE, A.-J. (1997): Multi-dimensional Time Support for Spatial Data Models, Research Report 253, May 1997. Department of Informatics, University of Oslo.

[12] SNODGRASS, R. (1995)(Eds.): The TSQL2 Language Design Committee. The TSQL2 Temporal Query Language. Kluwer.

[13] SNODGRASS, R. and AHN, I. (1985): A Taxonomy of Time in Databases. In: Proceedings of ACM SIGMOD International Conference on Management of Data. 236-246.

[14] STREIT (1998): Vorlesung Geoinformatik online. http://castafiore.unimuenster.de/vorlesungen/geoinformatik/index.html

[15] TANSEL *et al*. (1993): Temporal Databases. Benjamin/Cummings Publishing.

[16] WACHOWICZ, M. (1999): Object-Oriented Design for Temporal GIS. Taylor and Francis. London.

[17] WORBOYS, M. F. (1992): Object-Oriented Models of Spatiotemporal Information, Proceedings of GIS/LIS '92 Annual Conference, San Jose, California, USA, American Society of Photogrammetry and Remote Sensing, Bethseda, MA, USA, ISBN 0944426905, pp. 825-834.

[18] WORBOYS, M. F. (1994): Object-oriented approaches to geo-referenced information, International Journal of Geographical Information Systems, 8(4), pp. 385-399.

[19] ZIPF, A. UND ARAS, H. (2001): Realisierung verteilter Geodatenserver mit der OpenGIS SFS für CORBA. In: GIS. 3/2001. GIS - Geo-Informations-Systeme. Zeitschrift für raumbezogene Information und Entscheidungen. Heidelberg. pp. 36-41.

[20] ZIPF, A. AND MALAKA, R. (2001): Developing "Location based Services" (LBS) for Tourism - The service providers view. In: Sheldon, P.J, Wöber, K.W. and Fesenmaier, D.R. (Eds.): Information and CommunicationTechnologies in Tourism 2001. (Proceedings of ENTER 2001, 8th. International Congress on Tourism and Communications Technologies in Tourism. Montreal, Cannada. April 24-27). Springer Computer Science. Wien. New York. pp 83-92.

[21] ZIPF AND KRÜGER (2001): Ein objektorientierter Framework für temporale 3D-Geodaten. AGIT 2001, Symposium für Angewandte Geographische Informationsverarbeitung, 04.-06. Juli 2001, Salzburg. Austria.

8 APPENDIX – EXAMPLES

1.) XML-based example for
object time-stamping

```
<_Feature type="Streets" refOrder=”1000002”/>

<validHistory>
    <id>1000001</id>
    <orders>
        <linearOrder>
            <id>1000002</id>
            <structures>
                <combine>
                    <determinateDiscreteInstant>
                        <id>10000003</id>
                    </determinateDiscreteInstant>
                    <_FeatureValue>
                        <name>Main Street</name>
                        <attrib2>xxx</attrib2>
                        <_Geometry xsi:type=" LineString">
                         <coordinates decimal="." cs="," ts=" ">
                            3476622,54 3476710.25 …
                         </coordinates>
                        </_Geometry>
                    </_FeatureValue>
                </combine>
                <combine>
                    <determinateDiscreteInterval>
                        <id>1000004</id>
                    </determinateDiscreteInterval>
                    <_FeatureValue>
                        <name>Main Street</name>
                        <attrib2>yyy</attrib2>
                        <_Geometry xsi:type=" LineString">
                         <coordinates decimal="." cs="," ts=" ">
                            3476479,83 3476710.25 …
                         </coordinates>
                        </_Geometry>
                    </FeatureValue>
                </combine>
            </structures>
        </linearOrder>
    </orders>
</validHistory>
```

2.) XML-based example for
attribute time-stamping

```
<_Feature type="Streets">
    <name>Main Street</name>
    <attrib2>xxx</attrib2>
    <_Geometry xsi:type=" LineString" refOrder=”1000002”/>
</_Feature>

<validHistory>
    <id>1000001</id>
    <orders>
        <linearOrder>
            <id>1000002</id>
            <structures>
                <combine>
                    <determinateDiscreteInstant>
                        <id>10000003</id>
                    </determinateDiscreteInstant>
                    <_GeometryValue>
                     <coordinates decimal="." cs="," ts=" ">
                            3476622,54 3476710.25 …
                     </coordinates>
                    </_GeometryValue>
                </combine>
                <combine>
                    <determinateDiscreteInterval>
                        <id>1000004</id>
                    </determinateDiscreteInterval>
                    <_GeometryValue>
                        <coordinates decimal="." cs="," ts=" ">
                            3476479,83 3476710.25 …
                        </coordinates>
                    </_GeometryValue>
                </combine>
            </structures>
        </linearOrder>
    </orders>
</validHistory>
```