ChronoGeoGraph: A Development Framework for Spatio-Temporal Databases

D. Gubiani

Milano, 13 April 2010
Design is a crucial phase in the development of spatio-temporal database and information systems.

Spatial and temporal requirements are indeed much more difficult to deal with than the usual requirements of conventional systems.
Spatial Aspects

- The modeling language must provide tools to manage:
  - complex multidimensional data with spatial features (shape, extension and location)
  - different types of spatial relation,
  - incompleteness and/or inaccuracy of spatial data
  - multiple representations of the same spatial data
  - besides object-based representation, field-based representation
Temporal Aspects - 1

- Conventional database and information systems only maintain information about the current state of the world
- Temporal databases can keep track of the evolution of the modeled domain
- The modeling language must manage:
  - different temporal dimensions (valid time, existence time, transaction time, event time, and availability time)
  - temporal relations
The **valid time** of a fact is the time when the fact is true in the modeled domain
- existence time is associated with objects

The **transaction time** of a fact is the time when the fact is current in the database

The **availability time** of a fact is the time interval during which the fact is known and believed correct by the information system the database belongs to

The **event time** of a fact consists of the occurrence times of the real-world events that respectively initiate and terminate the valid time interval of the fact
Temporal Dimensions - 2

Andy is owner: 20 August 2006 to 23 February 2007

Tony is owner: 23 February 2007 to now

15 June 2006 to 15 January 2007

1 October 2006 to 28 February 2007

25 August 2006 to 23 February 2007
There exist many conceptual models for the management of temporal information or spatial information
- spatial: CONGOO, GeoER, GeoUML, GISER, OMT-G, GeoOMT, GeoFrame...
- temporal: TERM, RAKE, MOTAR, TEER, ITDM, STEER, ERT, TER...

Only a few of them deal with spatio-temporal information in an integrated and comprehensive way:
- STER
- STUSM
- MADS
- ST UML
- STED
### Comparative Analysis

<table>
<thead>
<tr>
<th></th>
<th>STER</th>
<th>ST USM</th>
<th>STUML</th>
<th>MADS</th>
<th>STED</th>
</tr>
</thead>
<tbody>
<tr>
<td>simplicity</td>
<td>++</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>expression power</td>
<td>+</td>
<td>+++</td>
<td>++++</td>
<td>++++</td>
<td>-</td>
</tr>
<tr>
<td>orthogonality</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>snapshot reducibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>upward-compatibility</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>nd</td>
</tr>
</tbody>
</table>
ChronoGeoGraph (CGG) is a spatio-temporal model that pairs the classical features of the EER model with a large set of spatial and temporal constructs.

Distinctive features:
- simplicity of the EER
- relevant properties of bidimensional spatial objects
- four distinct temporal dimensions
Besides basic entities CGG supports:

- **spatial entity**: characterized by a set of descriptive attributes plus a geometry of a given spatial data type
- **temporal entity**: entity provided with one or more temporal dimensions (existence, transaction, event, and availability times) and possibly with time-varying features
- **spatio-temporal entity**: obtained by merging spatial and temporal features
SPATIAL DATA TYPES

- **POINT**
- **LINE**
- **POLYGON**
- **MULTI POINT**
- **MULTI LINE**
- **MULTI POLYGON**
- **COLLECTION**
CGG borrows from EER the basic set of attribute types (simple, composite, optional, multivalued, and derived) and the possibility of constraining their cardinality.

*Spatial attributes* take their value over a spatial data type.

*Temporal attributes* are characterized by one or more temporal dimensions (valid, transaction, event and availability times).
Temporal attributes and collections

- Temporal dimensions can also be associated with the geometry of a spatial entity to model its changes over time (time-varying spatial entities).
- CGG distinguishes between *snapshot* and *lifespan cardinality constraints* on temporal attributes.
- *Temporal collection* is a set of attributes with a common temporal annotation that change in a synchronous way.
Besides basic relations, CGG features:
- spatial relations
- temporal relations
- synchronization relations

Moreover, CGG includes various types of specializations
**Spatial Relations**

*Spatial relations* are relations among spatial entities that impose spatial constraints between the geometries of the participating entities:

- topological, metric, and direction relations
- spatial aggregation

![Diagram showing spatial relations](image)
Synchronization relations are binary relations between temporal entities, provided with one or more temporal dimensions, that impose temporal constraints on their existence intervals.
Temporal relations are relations provided with one or more temporal dimensions (valid time, transaction time, event time, and availability time) or that have temporal features.

As in the case of temporal attributes, CGG distinguishes between *snapshot and lifespan cardinality constraints*. 

![Diagram showing Temporal Relations](image)
**Specialization and Cartographic Specialization**

- CGG borrows from EER the relation of specialization
  - disjoint or overlapping
  - total or partial

- The relation of *cartographic specialization* allows one to model different representations of the same spatial entity
  - with shape variation
  - with scale variation

![Diagram of City, Point, and Polygon relations](image-url)
TERRITORY SCHEMA

- A territory schema can be viewed as a degenerate purely spatial entity which admits one instance only.
- All spatial elements of a schema instance must be included in the instance of the territory schema.
- More restrictive conditions can be explicitly forced by using CGG spatial relations.
- It can be temporal.
**Fields**

- *Field* are features that vary over space
- They can be associated with either the whole schema territory or a single spatial entity
- They are characterized by a specific sampling type
  - regular/irregular points, isolines, regular/irregular cells, TIN
- Spatial interpolation functions over fields can be defined
- The temporal dimensions can be associated with fields
• CGG explicitly keeps track of the events that change the state of a relevant element
• Additional information about the nature and the effects of an event can be associated with the edge that connects the event to the element on which it acts

![Diagram of an event associated with a plot](image-url)
Case Studies - Contaminated Sites
Case Studies - Sanity
Different applications need to deal with spatial data under:
- different viewpoints
- different geometric forms
- different degrees of precision

Two fundamental granularity dimensions:
- spatial granularity: variations of the geometry of the objects
- object granularity: variations of the set of domain objects
Existing Conceptual Models

- Existing conceptual models that allow one to deal with multiple spatial representations
  - MADS, OMT-G, ST-UMS, Perceptory

- A common limitation: they provide no consistency constraints on multiple data representations. That is, there is no way to constrain the topological relations connecting spatial objects with multiple representations to be consistent.
**Topological Consistency**

- Different solutions were proposed in the context of multiresolution geographic maps.

- Given two differently-grained maps which have some spatial objects in common, it must be checked whether they are consistent and, if not, whether they are at least similar.

- Belussi, Catania and Podestà (2005) formalize the notions of consistency and similarity of multiresolution maps, with respect to topological information, taking into account possible changes in the spatial dimension:
  - equality-based consistency
  - distance-based consistency
How to extend the CGG model to deal with multiple representations associated with the same spatial object?

- resolution
- cartographic specialization
- multiple representations of topological relations
- spatial aggregation
Resolution: the minimum distance at which two points must be to be distinguishable.
Cartographic Specialization: different representations can be associated to an entity to model
- point of view changes
- shape changes
- resolution changes
There can be situations where there exists no a single topological relation which can be associated with a given pair of instances in all representations.
• Associating different topological relations with different spatial representations do not prevent inconsistencies from being generation at instance level.
CGG allows multiple representations of topological relations, each one relating pairs of (representations of) spatial entities at the same resolution.
• Representations must satisfy distance-based consistency requirements for CGG topological relations.

<table>
<thead>
<tr>
<th>L-P</th>
<th>c</th>
<th>in</th>
<th>d</th>
<th>cv</th>
<th>cvb</th>
<th>eq</th>
<th>ovi</th>
<th>tc</th>
<th>min</th>
</tr>
</thead>
<tbody>
<tr>
<td>in</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>d</td>
<td>4</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>cvb</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>on</td>
<td>6</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>ovd</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>ovi</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>6</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>tc</td>
<td>6</td>
<td>5</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>min</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Besides the standard one, CGG includes a weakened form of spatial aggregation, which may relax the covering constraint in case of multiple representations.
A Case Study - Integration of the Three Schemas
We developed a JAVA tool to synthesize ChronoGeoGraph diagrams

DISTINCTIVE FEATURES:
- it includes all CGG constructs
- it includes dialog windows
- it includes popup menus
- it automatically checks integrity constraints
- it includes two translation modules into logical models (relational schema and XML)
We proposed a relational model with four temporal dimensions of CGG.

We defined a translation of CGG schemas into the proposed temporal model.
A Relational Model with Four Temporal Dimensions

- It uses of tuple timestamping, to preserve 1NF
- It makes use of time intervals to model valid, transaction, and availability times and time instants to model event time
- It defines temporal keys as suitable temporal extensions of the primary keys of the original atemporal relations
- Temporal schemas (resp., instances) are partitioned into a current component and a historical one
- A number of constraints guarantee the consistency of the values of the different temporal dimensions
A temporal schema with the four temporal dimensions can be obtained by an atemporal schema \( R(X) \) as follows:

1. **Adding Valid Time**
   
   \[
   R(X, VT_{\text{start}}, VT_{\text{end}})
   \]

2. **Adding Event Time**
   
   \[
   R(X, VT_{\text{start}}, VT_{\text{end}}, ET_{\text{start}}, ET_{\text{end}})
   \]

3. **Adding Transaction Time**
   
   \[
   R(X, VT_{\text{start}}, VT_{\text{end}}, ET_{\text{start}}, ET_{\text{end}}, TT_{\text{start}})
   \]
   
   \[
   R\_history(X, VT_{\text{start}}, VT_{\text{end}}, ET_{\text{start}}, ET_{\text{end}}, TT_{\text{start}}, TT_{\text{end}})
   \]

4. **Adding Available Time**
   
   \[
   R(X, VT_{\text{start}}, VT_{\text{end}}, ET_{\text{start}}, ET_{\text{end}}, TT_{\text{start}}, AT_{\text{start}}, AT_{\text{end}})
   \]
   
   \[
   R\_history(X, VT_{\text{start}}, VT_{\text{end}}, ET_{\text{start}}, ET_{\text{end}}, TT_{\text{start}}, TT_{\text{end}}, AT_{\text{start}}, AT_{\text{end}})
   \]
Temporal Constraints: Some Examples

- Constraints for availability time:
  - $\exists t \in T^g \quad t = AT_{\text{start}}$
  - $AT_{\text{end}} \leq \text{now} \lor AT_{\text{end}} = uc$
  - $AT_{\text{start}} < AT_{\text{end}}$

- Constraints between availability and transaction times:
  - $AT_{\text{start}} \leq TT_{\text{start}}$
  - $AT_{\text{end}} \leq TT_{\text{end}}$
  - $AT_{\text{end}} = uc \Rightarrow TT_{\text{end}} = uc$
  - $TT_{\text{end}} = uc \Rightarrow AT_{\text{end}} = uc$
  - $AT_{\text{end}} \leq TT_{\text{start}} \Leftrightarrow TT_{\text{start}} = TT_{\text{end}}$
Both problems have been already addressed in the temporal databases literature, but there are no consensus solutions.

**Our goal:** to guarantee an appropriate trade-off between expressiveness and effectiveness.

- to maintain the notion of temporal key and temporal dependency as simple and easy to manage as possible.

We define TFDs as temporal generalizations of (atemporal) functional dependencies (FDs).
Temporal Functional Dependencies

Definition (Temporal Functional Dependencies)

Given a temporal relation $R$ with atemporal schema $R(\mathbf{X})$ and a TFD $Z \rightarrow_{T} Y$, with $Z, Y \subseteq \mathbf{X}$, we say that an instance $r \in R$ satisfies the TFD if and only if, for each pair of tuples $a, b \in r$, if $a[Z] = b[Z]$, $\pi_{VT}(a) \cap \pi_{VT}(b) \neq \emptyset$ (their valid time intervals overlap), and $\pi_{TT}(a) \cap \pi_{TT}(b) \neq \emptyset \lor \pi_{AT}(a) \cap \pi_{AT}(b) \neq \emptyset$ (their transaction or availability time intervals overlap), then $a[Y] = b[Y]$. 
Temporal Keys for Temporal Relations

<table>
<thead>
<tr>
<th>Cases</th>
<th>Temporal keys</th>
<th>Temporal dimensions</th>
</tr>
</thead>
<tbody>
<tr>
<td>atemporal</td>
<td>$R(K, ...)$</td>
<td>-</td>
</tr>
<tr>
<td>valid time</td>
<td>$R(K, VT_start, ...)$</td>
<td>$V$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$VE$</td>
</tr>
<tr>
<td>transaction time</td>
<td>$R(K, ...)$</td>
<td>$T$</td>
</tr>
<tr>
<td></td>
<td>$R_history(K, TT_start, ...)$</td>
<td>$TA$</td>
</tr>
<tr>
<td>valid and transaction times</td>
<td>$R(K, VT_start, ...)$</td>
<td>$VT$</td>
</tr>
<tr>
<td>transaction times</td>
<td>$R_history(K, VT_start, TT_start, ...)$</td>
<td>$VTE$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$VTA$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$VTAE$</td>
</tr>
</tbody>
</table>
Does the Key Constraint Suffice with an Interval-Based Representation

All possible choices for the attributes of the temporal key do not detect the inconsistency

- \((SSN, VT\_start), (SSN, VT\_end),\) and \((SSN, VT\_start, VT\_end)\)

<table>
<thead>
<tr>
<th>SSN</th>
<th>Salary</th>
<th>VT_start</th>
<th>VT_end</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXXNNNN88HH</td>
<td>1000</td>
<td>15/10/2000</td>
<td>31/07/2006</td>
</tr>
<tr>
<td>XXXNNNN88HH</td>
<td>1200</td>
<td>01/10/2003</td>
<td>31/07/2007</td>
</tr>
</tbody>
</table>

Temporal consistency must be explicitly checked
Temporal Consistency Constraints

- Constraint checking can be confined to insertions in the current database
  - with respect to current tuples:

  \[ \forall a, b \in R(X) \forall Y \subseteq X (a[K] = b[K] \land \pi_{VT}(a) \cap \pi_{VT}(b) \neq \emptyset \rightarrow a[Y] = b[Y]) \]

  - with respect to historical tuples:

  \[ \forall a \in R(X) \forall b \in R_{-history}(X) \forall Y \subseteq X (a[K] = b[K] \land \pi_{VT}(a) \cap \pi_{VT}(b) \neq \emptyset \land \pi_{AT}(a) \cap \pi_{AT}(b) \neq \emptyset \rightarrow a[Y] = b[Y]) \]
We defined a translation of CGG schemas into the proposed temporal model.

The translation introduces a set of relation schemas for every temporal entity and relation:
- a root schema, called kernel, that plays the role of reference schema for all relation schemas generated by a given entity or relation.
- a relation for each property (temporality, geometry or attributes) of the given entity or relation.
- each single relation schema is linked to the kernel by means of a suitable foreign key.
An Example

Person

Person_kernel(SSN)
Person_atemporal(SSN, firstName, lastName)
Person_work(SSN, VT_start, VT_end, TT_start, work)
Person_work_history(SSN, VT_start, VT_end, TT_start, TT_end, work)
A Case Study - 1

Diagram showing relationships between Commune, ASL, Italy, Patient, codeISTAT, contain, reside, id, type, admission, firstName, and lastName.
A Case Study - 2

Introduction
ChronoGeoGraph Model
Software Tool and Translations
Comparison of CGG and MADS

Software Tool
Translation into Relational Model
Translation into XML Schema

A Case Study - 2

D. Gubiani
ChronoGeoGraph 49
A Case Study - 2

```
-- Update of a current tuple with transaction time
CREATE OR REPLACE TRIGGER Patient_temporality_updateTT
    BEFORE UPDATE ON Patient_temporality
    FOR EACH ROW
DECLARE
    now timestamp;
BEGIN
    SELECT systimestamp(0) INTO now FROM dual;
    INSERT INTO Patient_temporality_history (SSN, VT_start, VT_end, TT_start, TT_end)
        VALUES (:old.SSN, :old.VT_start, :old.VT_end, :old.TT_start, now);
    :new.TT_start := now;
END;
```

```
CREATE OR REPLACE TRIGGER Contain_insert
    BEFORE INSERT OR UPDATE ON contain_t
    FOR EACH ROW
BEGIN
    IF (Contain_find_geomViolations(:new.codeISTAT, :new.id, :new.VT_start, :new.VT_end) > 0)
        THEN raise_application_error(-20001, 'Participating entities violate the spatial constraint!');
    END IF;
END;
```
Beside a translation into a temporal extension of the relational model, we proposed a

**translation into XML Schema**

**Why XML/XML Schema?**

- The mapping of CGG schemas into XML ones support the exchange and integration of spatio-temporal data among different applications
- Since an XML schema is an XML document, one can take advantage of XML query languages to query XML schema
- Since both the schema and its instance are stored as XML documents, a single query can access both the schema and instance in the same query
Encoding of basic EER is straightforward:

- entity E $\rightarrow$ complexType element E
- attributes of E $\rightarrow$ sequence in complexType element E
- key attribute $\rightarrow$ key element
- relation $\rightarrow$ subelement or keyref element
Translation of EER Features - 2

```xml
<xs:element name="Person" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="SSN" type="xs:string" minOccurs="1" maxOccurs="1" />
      <xs:element name="name" minOccurs="1" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="lastName" type="xs:string" minOccurs="1" maxOccurs="1" />
            <xs:element name="firstName" type="xs:string" minOccurs="1" maxOccurs="1" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:complexType>
```
Spatio-Temporal Translation

- It exploits the main features of XML Schema:
  - it takes advantage of GML for spatial features
  - it uses timestamp attributes for temporal dimensions

- However, XML Schema cannot encode and check all spatio-temporal constraints
  Solution: to add annotations as appinfo
<xs:element name="geometry">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gml:Point" />
    </xs:sequence>
  </xs:complexType>
</xs:element>
Translation of Spatio-Temporal Entities - 2

```xml
<xsd:element name="type" minOccurs="1" maxOccurs="unbounded">
  <xsd:annotation>
    <xsd:appinfo>
      <minOccursLS>1</minOccursLS>
      <maxOccursSS>1</maxOccursSS>
    </xsd:appinfo>
  </xsd:annotation>
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="vti" type="xsd:date" use="required"/>
        <xsd:attribute name="vtf" type="xsd:date"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
```
Translation of Spatio-Temporal Entities - 3

```xml
<xsl:element name="lifespan" minOccurs="1" maxOccurs="unbounded">
  <xsl:annotation>
    <xsl:appinfo>
      <minOccursLS>1</minOccursLS>
      <maxOccursSS>1</maxOccursSS>
    </xsl:appinfo>
  </xsl:annotation>
  <xsl:complexType>
    <xsl:simpleContent>
      <xsl:extension base="xsl:string">  
        <xsl:attribute name="lsi" type="xsl:date" use="required" />
        <xsl:attribute name="lsf" type="xsl:date" />
      </xsl:extension>
    </xsl:simpleContent>
  </xsl:complexType>
</xsl:element>
```
Translation of Spatio-Temporal Relations

```xml
<xsl:element name="Commune" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="contains" minOccurs="0" maxOccurs="unbounded">
        <xs:annotation>
          <xs:appinfo>
            <typet>contains</typet>
            <prTopological>First</prTopological>
          </xs:appinfo>
        </xs:annotation>
        <xs:complexType>
          <xs:sequence>
            <xs:element name="SanityStructure">
              ...
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xsl:element>
```
The constraints to be checked are of three types:
- basic constrains
- W3C XML Schema Language constrains
- advanced spatio-temporal constraints
**Introduction**

**ChronoGeoGraph Model**

**Software Tool and Translations**

**Comparison of CGG and MADS**

**Spatio-Temporal Validator**
A Case Study - 1
A Case Study - 2
A Case Study - 3

...<SanityStructure>
   <geometry>
      <gml:Point>
         <gml:pos>
            <gml:X>4</gml:X>
            <gml:Y>4</gml:Y>
         </gml:pos>
      </gml:Point>
   </geometry>
   <id>SS8</id>
   <type tti="2001-01-01" vti="2001-01-01">hospital</type>
   <public><lifespan lsi="2001-01-01"/></public>
</SanityStructure>
...

...<Patient>
   <reside tti="2001-01-01" vti="2001-01-01"/>
   <Commune keyref="COM2"/>
   </reside>
   <SSN>888888888<SSN>
   <name>
      <lastName>Bellucci</lastName>
      <firstName>Monica</firstName>
   </name>
   <lifespan lsi="1965-04-18" lsf="1956-12-03" eti="2001-01-01" etf="2001-01-01"/>
</Patient>
...

Error Entity SanityStructure SS8 The Lifespan time interval is not correct
Error Entity SanityStructure SS8 The schema territory does not contain this instance of the entity
Error Entity Patient BLMNC The Lifespan time interval is not correct
MADS, proposed in 1995 by Spaccapietra’s research group as an extension of Entity-Relationship model, is become an object-oriented conceptual model. It allows to model thematic, spatial and temporal data. It supports multiple representations.
An Example
Classical Features

Some differences depend on basic concepts

<table>
<thead>
<tr>
<th>feature</th>
<th>CGG</th>
<th>MADS</th>
</tr>
</thead>
<tbody>
<tr>
<td>entities of real world</td>
<td>entities (attributes)</td>
<td>objects (attributes and methods)</td>
</tr>
<tr>
<td>identifier</td>
<td>system-defined attribute (oid)</td>
<td>key attributes required</td>
</tr>
<tr>
<td>weak entities</td>
<td>adding the role of the identifying relationship</td>
<td>ER representation</td>
</tr>
</tbody>
</table>
# Temporal Features

<table>
<thead>
<tr>
<th>feature</th>
<th>CGG</th>
<th>MADS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Temporal dimensions</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- valid/existence time</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>(state diagrams)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- transaction time</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>- event time</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>- availability time</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>synchronization relations</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>temporal collection</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Spatial Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>CGG</th>
<th>MADS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object-based view</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Field-base view</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Geometry</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- object/entity</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- attribute</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- relations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Spatial constraining relations</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- topological</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- metrical</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- directional</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>- aggregation</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Territory schema</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
CGG: resolution and topological consistency  
VS  
MADS: perceptions
MADS is very flexible but the graphical representation is not complete
  - it allows to add several constraints
  - it becomes more complex

One of the main goals of CGG is to be simple
  - the graphical representation is complete
  - it is less flexible than MADS


CGG Project

http://dbms.dimi.uniud.it/cgg/