

Script-based Presentation of Simulation Results

Roland Jesse*

Abstract

The presentation of simulation results often targets a wide range of audience. An appropriate parameterisation is therefore needed for an automatic presentation of any visualisation as well as an interaction based examination of simulated data. This paper presents an XML Schema named TIGEOP that helps to fulfil this parameterisation task. The practicability of this approach is shown at the example of climate data.

1 Introduction

The examination of various simulated scenarios is deemed as a valuable option for play-through scenarios in a variety of areas. An example is the use of simulated weather data in agriculture and catastrophic prevention programs. Simulated data is often used instead of real data as it provides for greater flexibility. Sometimes, real data is simply not available at all or not in suitable quantities sufficient for planned investigation purposes. At other times, only part of the required data granularity is available whereas the remaining data needs to be gained by simulation.

Presentation of simulation results is often challenged by varying motivations and expectations of targeted users. Generic visualisations do not necessarily meet any user's target function regarding the analysis of generated data. Furthermore, they might require temporal control to some degree. This might be reasoned in the temporal nature of the presentation data itself, such as in the case of climate data. Further motivation might result of a user's need for time-varying presentation or the consideration of user-input handling.

Controlling the visual representation of a simulator's output on a script basis helps to provide for pre-modelled and user oriented presentation including consideration of any possible temporal constraints. Thereby, the set of presentation variables and their respective mapping is to be described following an application independent specification. This allows for simulation output to be presented depending on domain constraints as well as user actions. Furthermore, a script-based presentation model allows to comply to temporal constraints as defined by the underlying format specification.

This work aims at the script-based presentation of simulation results with special emphasis on generality and the evaluation of temporal constraints. The practicability of this approach is shown at the example of climate data.

Some details about the example data used throughout this paper are presented in Section 2. An architecture for the generation of presentations in the context of the work presented here is shown in Section 3. This is followed by Section 4 defining a temporal algebra as formal basis for temporal constraints of the XML Schema definition as presented in Section 5.

*School of Computer Science, Department of Simulation and Graphics, 39106 Magdeburg, Germany

This work is concluded in Section 6. Planned extensions to this work are presented there as well.

2 Simulation Data

Possible interest in climate data spans a wide range of potential analysis areas. A common approach is the investigation of a curvilinear grid defined by longitude, latitude, and altitude. This allows to compare different area's data and is useful for weather forecasts. Another approach is to investigate data for a specific location only by sampling data at concrete location points. This preserves a reasonable amount of data and allows to examine data spanning a more extensive time frame. This latter case forms the basis of the work presented here. Precisely, daily weather data are used.

2.1 The Tool

As available weather databases provide incomplete daily data or only monthly summaries, a simulator is used to generate synthetic daily climate data that provides the same statistical characteristics as real data for the desired location. Different models have been proposed for this purpose. For the context of this work, the ClimGen¹ simulator is used. It is based on the WGEN model by Richardson and Wright (1984) but generates more correct daily weather data out of given monthly data summaries. The ClimGen tool allows for the generation of data containing total solar radiation, maximum and minimum temperature, rainfall, and wind-run.

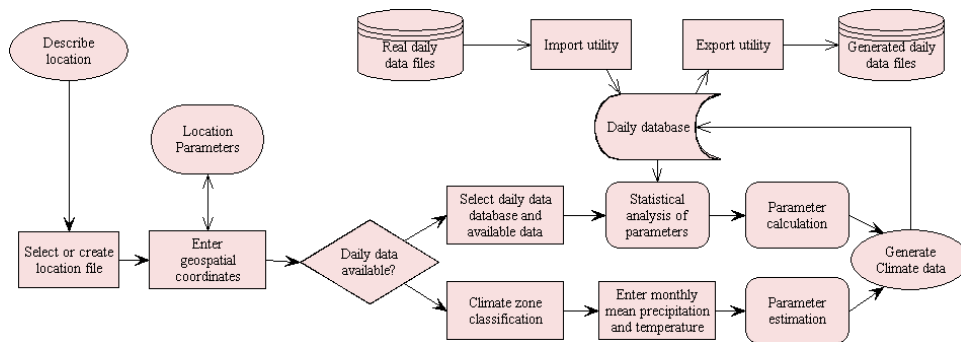


Figure 1: Climgen flow chart. This figure is part of the ClimGen user manual and © by Roger Nelson.

¹More information about ClimGen, its availability, and a comparison compendium of ClimGen and other climate data generators is available at <http://c100.bsyse.wsu.edu/climgen/>.

2.2 Input

In order to generate any reliable data, the simulator needs to be parameterised accordingly. Figure 1 shows the principle data flow of the ClimGen simulator and documents in which regard the parameters influence the simulator's output. The parameterisation is done by describing a *location* and specifying a set of *location parameters*. As an exemplary location, the place Magdeburg (Germany) is used. The location Magdeburg is described by a latitude of 52.13° , a longitude of 11.62° , and an elevation of 79.00 m . ClimGen allows various degrees of input data completeness. Eventually missing parameters are estimated by the program. The following table lists the *monthly mean precipitation and temperature* values that are used as basis for data generation:

month	$mean(T_{min})$	$mean(T_{max})$	$mean(prec.)$
January	$-1.8^\circ C$	$13.0^\circ C$	34.8 mm
February	$-0.6^\circ C$	$12.0^\circ C$	30.0 mm
March	$0.6^\circ C$	$13.0^\circ C$	35.0 mm
April	$4.6^\circ C$	$16.9^\circ C$	39.2 mm
May	$6.1^\circ C$	$37.1^\circ C$	47.6 mm
June	$14.3^\circ C$	$24.3^\circ C$	63.8 mm
July	$13.0^\circ C$	$29.0^\circ C$	58.4 mm
August	$12.4^\circ C$	$24.5^\circ C$	52.3 mm
September	$10.5^\circ C$	$20.5^\circ C$	38.1 mm
October	$5.4^\circ C$	$15.7^\circ C$	33.8 mm
November	$-5.6^\circ C$	$10.9^\circ C$	37.3 mm
December	$-7.0^\circ C$	$11.2^\circ C$	39.2 mm

2.3 Output

Using the input as described above, ClimGen generates usable data about precipitation and temperature. As some parameters were only estimated, possibly generated data about solar radiation and wind-speed are not of reliable nature and therefore not used.

The data is written in files with one file for each year output is generated for. These files are organised in a tabular layout with one line per daily data. The generated data is put into respective columns. The following table shows an exemplary excerpt from the generated output, precisely data for January 10 to 20, 1999:

Day of year	Precipitation	T_{max}	T_{min}
⋮	⋮	⋮	⋮
10	0.00	3.32	1.69
11	0.00	0.99	-0.01
12	0.00	2.82	0.39
13	0.00	1.16	0.13
14	3.74	0.88	-4.36
15	0.00	8.13	4.62
16	0.00	4.56	3.56
17	0.00	3.24	1.91
18	0.00	5.50	1.83
19	0.00	7.88	1.41
20	0.00	9.37	0.96
⋮	⋮	⋮	⋮

3 Architecture

Figure 2 shows a client/server architecture to be used for the generation of presentations controlled by a script description. The client is used as the interface to the user. It is responsible for the implementation of individual presentation techniques. The server holds responsible for the mapping of geometry as well as the (semi-automatically) construction of an XML script controlling temporal operation of all presentation styles. Both—client and server—do not necessarily need to reside on the same physical computer. Communication is done by use of a TCP/IP-based protocol.

In the figure, hexagons represent active architecture components whereas storing components (be it databases, scripts, or internal data structures) are represented by rounded boxes. The double-arrows indicate flow of physical data (files, scripts, database content) whereas single-arrows are used for flow of internal data.

The application dependent server component is the geometry mapper. It is the source of geometry definition and therefore providing the overall scene context. The design of this component is formally based on the work presented by Kreuzeler and Schumann (2002). Their model was developed in the context of Visual Data Mining. But its general nature allows for applications in other domains as well. In principle, the model is based on the definition of *information objects* IO_i that combine to an *information space* $\mathbf{IM} = \{IO_1, \dots, IO_n\}$ with $IO_i = IO_j \iff i = j$ and $i, j, n \in \mathbb{N}$. Each information object represents some real world data. In order to parameterise these objects according to their represented data characteristics, an attribute function *attr* is provided: $\mathbf{AM} = attr(\{IO_1, IO_2, \dots, IO_n\}) = \{A_1, A_2, \dots, A_k\}$ with $A_i = A_j \iff i = j$ and $i, j, k, n \in \mathbb{N}$. Thereby, \mathbf{AM} is the *attribute set* of the respective information objects. For the purpose of defining relations between IO_i the *information structure* \mathbf{IS} is introduced as $\mathbf{IS} \subseteq \mathbf{IM} \times \mathbf{IM}$.

Specific model instances for the mapper are to be constructed semi-automatically. For the purpose of presenting simulation results, the information space is generated by executing

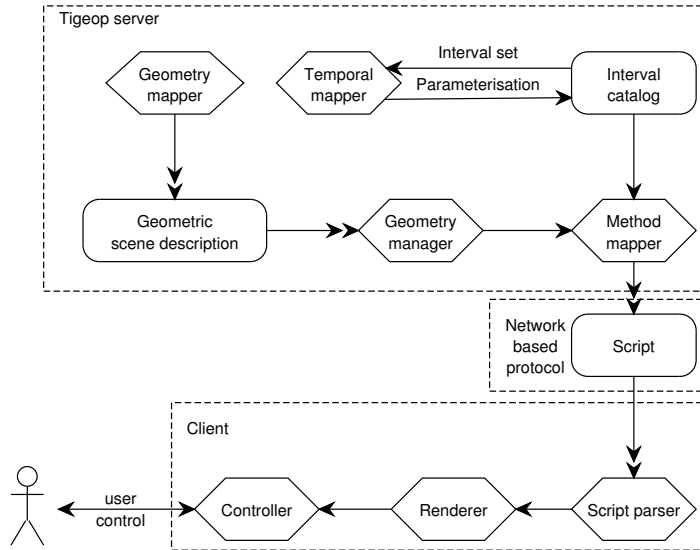


Figure 2: Overview of the TIGEOP-based client/server architecture for presentation systems.

a user defined visualisation pipeline. Creation of any possible objects in this space as well as the setup and modification of attribute sets is based on a user controlled presentation description as outlined in the following two sections.

4 Temporal Relation Algebra

Considering temporal constraints for the modelling of a presentation of simulation results helps to convey structural information about simulated data as well as intrinsic data characteristics. The script generation architecture as outlined in the previous section includes a *temporal mapper* component. This mapper basically evaluates a temporal algebra for the purpose of generating and updating an interval catalogue to be used for presentation parameterisation. Continuing the work done by Jesse et al. (2002), this section presents the specification of the formal algebra that is used as the basis for temporal control of different presentations.

4.1 Time Granularity

Allen (1991) outlines different ways of modelling time representations. For the purpose of controlling presentation methods, a combination of absolute dates and duration specification is used in the context of this work. This allows for the generation of a complete linear

order of all events as well as the generation of temporal networks with a fine grained model granularity in accordance with Bettini et al. (2000).

In contrast to the time model based on hierarchical granularities as presented by Merlo et al. (1999), the model in this work uses a complete description of temporal events that is based on XML Schema's `date` data type and spans all individual granularities. This provides for the following advantages:

- Inherited temporal types are not needed as the `date` type allows to model temporal entities spanning the *complete* time hierarchy as presented by Merlo et al.
- No coercion function is needed for the mapping of entities in one granularity to entities in other (superclass) granularities.
- The handline of different granularities is implicitly given.

In order to model any event and all event's identity, one only needs to specify a starting point t_0 and a duration d , respectively. This implies complete ordering information.

4.2 Algebra Specification

In order to name the different relation sets used for the algebra, some variables are introduced. The parameter set for a specific presentation is denoted as δ . All defined presentation styles (methods) are represented by \mathcal{M} . The available interval set is named \mathcal{I} whereas an individual interval is labelled I . The set of time stamped events is described by E . The functions τ and ψ are used for operations on events and intervals. Discretisation of an interval resulting in an interval set is accomplished by τ and the mapping of any specific event onto its respective interval results from ψ .

The specification of the temporal model is given as tuple $SPEC = \langle S, OP, REL, X, REQ \rangle$. The elements of S represent all *sorts (types)*. S^* names the set of all sort chains. For each $\omega \in S^*$ and each $s \in S$, the set of all possible operators is named $OP_{\omega,s}$. The elements op of $OP_{\omega,s}$ are therefore projections $op : S^* \times S \rightarrow S$. For each $\omega \in S^*$ represents REL_{ω} the set of all possible relations. The set $\langle S, OP, REL \rangle$ is commonly referred to as signature of the specification. X marks the defined variable set. Last, but not least, names $REQ \subset F_{SIG}(X)$ the set of formulas that define the algebra of the specification.

To the benefit of readability this syntax is simplified to some degree. The different tuple components are presented in a tabular form whereby each set is prefaced with a keyword. The sorts are presented after the keyword *sorts*, the operations follow *opns*, relations are listed after *rels*. All variables are prefaced by *vars* and the formulas of the algebra follow *reqs*.

Presentation parameterisation $\delta(\mathcal{M}, \mathcal{I}, \psi, \tau)$

sorts: $I, \mathcal{I}, E, \mathcal{M}, \mathcal{W}_{rep}$

opns: $\psi : E \rightarrow I$

$\tau : I \rightarrow \mathcal{I}$

	$+ : \mathcal{I} \rightarrow \mathcal{I}$
rels:	$<, =, > : \mathcal{I} \mathcal{I}$ $\subseteq, \subseteq, = : \mathcal{I} \mathcal{I}$ $\neq : \mathcal{M} \mathcal{M}$ $\neq : \mathcal{W}_{rep} \mathcal{W}_{rep}$
vars:	$e_1, \dots, e_n : E$ $I_1, \dots, I_n : \mathcal{I}$ $m_1, \dots, m_n : \mathcal{M}$ $world_{rep}^1, \dots, world_{rep}^n : \mathcal{W}_{rep}$
reqs:	$\psi(e_n) < \psi(e_m) \iff e_n < e_m$ $\tau : \forall I_i \in \mathcal{I} \implies I_1^b \leq I_i^b \wedge I_n^e \geq I_i^e$ (limiting interval borders) $I_a < I_b \iff I_a^b \geq I_b^b \wedge I_a^e < I_b^e$ (limiting interval lengths) $I_a = I_b \iff I_a^b = I_b^b \wedge I_a^e = I_b^e$ $I_a > I_b \iff I_a^b \leq I_b^b \wedge I_a^e > I_b^e$ $\mathcal{I} + \lambda = \mathcal{I}$ $\mathcal{I}_a = \{I_1, \dots, I_a\} \wedge \mathcal{I}_b = \{I_{a+1}, \dots, I_b\} \implies \mathcal{I}_a + \mathcal{I}_b = \{I_1, \dots, I_b\}$ (interval continuity) $\mathcal{I}_a \subseteq \mathcal{I}_b \wedge \mathcal{I}_b \subseteq \mathcal{I}_c \implies \mathcal{I}_a \subseteq \mathcal{I}_c$ (transitivity) $m_a \neq m_b \iff world_{rep}^a \neq world_{rep}^b$ (seclusiveness of styles)

Table 1: Specification of a presentation parameterisation.

Table 1 presents the specification of the temporal presentation parameterisation regarding different presentations. For each predefined or otherwise determined interval, a combination of presentation methods ($m_i \in \mathcal{M}$) can be defined and parameterised. The intervals determining a presentation are both—complete and self-contained. That is, any time stamp (event) can be mapped onto an interval (by use of ψ) and no interval of the model exceeds the interval set for the overall presentation.

5 Schema for presentation description

This work presents the use of XML Schema² as a formal basis for the script-based presentation description. By using formally described XML documents, easy reuse of presentation scripts is ensured as well as automated script verification. XML furthermore allows for streaming support, object orientation, and therefore generality in appliance for a maximum number of application domains. This schema is named TIGEOP which is an abbreviation for *temporally influenced geometry presentation*. It not only aims at presenting simulation results for the purpose of climate data visualisation but supports the presentation of any geometry. The *server* components of the architecture outlined in section 3 hold responsible for the context definition. Examples of potential applications besides the work presented here are: a workbench for information fusion such as the one presented by Dunemann et al.

²Information about XML Schema, according tools, and documentation is available at <http://www.w3.org/XML/Schema>.

(2002) and the use of hybrid rendering styles for presentation as outlined by Jesse and Isenberg (2003). A generic architecture for the creation and evaluation of such scripts was introduced in Section 3. Section 5.1 discusses the overall structure of the script documents, whereas Section 5.2 provides an illustrating example.

5.1 Script Structure

The presentation script requirements as discussed so far are combined in a formal XML Schema definition. Thereby, the model entities of the framework by Kreuzeler and Schumann (2002) responsible for the geometry mapping are enriched by temporal presentation aspects according to the aforementioned temporal algebra. All resulting entities are mapped onto respective XML elements to form the schema definition. The schema is presented here completely in order to allow for easy reuse and verification of the model basis. As it exclusively uses the XML Schema facilities as defined by the W3C its content structure follows a well defined standard and is of self-documenting nature.

Listing 1 shows the schema definition. Individual types of the schema represent direct mappings of the formal notations presented in Sections 3 and 4. Lines 11–17 of the listing define the overall structure of a document instance. A document consists of a set of presentation methods (rendering styles), the geometric scene description, and a sequence of object presentation definitions. The *string* type for the geometry definition in line 14 is to be interpreted as containing a scene graph in Open Inventor format.

Presentation methods are defined in lines 18–33 and bridge between the attribute set **AM** of Kreuzeler and Schumann’s framework as outlined in Section 3 and the set \mathcal{M} of the algebra from Section 4. Attributes for the individual methods are defined as *methodProperties* in lines 30–33. Their content model is empty. Therefore the *property* element for each *presentation method* conveys both the name and the value of a specific method attribute. Possible interpretations of the content of these attributes in a document instance include the rendering to producing either photorealistic or non-photorealistic (NPR) image components.

The heart of a document in regard to the temporal modelling of a visualisation is the definition of object presentations in lines 34–48. These presentations represent the evaluation of individual rendering styles and accompanying parameters at defined time points for the given durations. The temporal dependencies conform to the above algebra definition. Similar to the method properties mentioned above, the content model for an object’s *present* description is empty which results in all parameters being specified as element attributes.

Listing 1: XML Schema definition for TIGEOP.

```

<?xml version='1.0' encoding='UTF-8'?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
5      Schema for temporally influenced geometry presentation.
      Copyright (c) 2002/3 University of Magdeburg. All rights
        reserved.
    </xsd:documentation>

```

```

</xsd:annotation>
<xsd:element name="tigeop" type="tigeopType"/>
10
<xsd:complexType name="tigeopType">
  <xsd:sequence>
    <xsd:element name="presentationmethods" type="
      presentationMethodType" minOccurs="0"/>
    <xsd:element name="ivscene" type="xsd:string" minOccurs="1"
      maxOccurs="1"/>
15    <xsd:element name="sceneobjects" type="sceneObjectsType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="presentationMethodType">
  <xsd:sequence>
20    <xsd:element name="method" type="methodType" maxOccurs="
      unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="methodType">
  <xsd:attribute name="id" type="xsd:integer"/>
25 <xsd:attribute name="name" type="xsd:string"/>
  <xsd:sequence>
    <xsd:element name="property" type="methodPropertyType"
      maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
30 <xsd:complexType name="methodPropertyType">
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="value" type="xsd:string"/>
</xsd:complexType>
<xsd:complexType name="sceneObjectsType">
35 <xsd:sequence>
  <xsd:element name="object" type="sceneObjectType" maxOccurs
    ="unbounded"/>
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="sceneObjectType">
40 <xsd:attribute name="id" type="xsd:string"/>
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:element name="present" type="presentObjectType"/>
</xsd:complexType>
<xsd:complexType name="presentObjectType">
45 <xsd:attribute name="method" type="xsd:integer"/>
  <xsd:attribute name="start" type="xsd:date"/>
  <xsd:attribute name="duration" type="xsd:date"/>
</xsd:complexType>
</xsd:schema>

```

5.2 Application

The use of an information mural (Jerding and Stasko, 1998) for visualisation of climate data is documented by Jesse (2002). Therein, the geometry mapping framework presented in section 3 is used as well. Creation of the information objects IO_i correlates directly to the available daily weather data as generated by ClimGen. That is, for each available day, an information object IO is created. The geometric shape of an object is a vertical line in 2-D respective a cuboid in 3-D. This allows to order all IO_i along the x axis. Other object characteristics as well as the mural restrictions are ensured by assigning appropriate attributes $A_i \in \mathbf{AM}$. These attributes include position, shape modification, and colour. The x position is determined by appropriately moving along the axis according to the overall coordinate maximum of the display. In contrast to the original mural implementation by Jerding and Stasko (1998), the case of overlapping pixels is not encoded in colour intensity but in scaling the lines vertically by a precomputed factor. Positioning along the y axis is done according to the temperature values of the respective day's data. The line bottom represents the temperature minimum value, the line's height represents the temperature span. Colour mapping is used to express the precipitation data in an information object. Figure 3 shows snapshots of the mural based on a TIGEOP document with its respective object presentation excerpt in Listing 2. Sub-figure 3(a) shows the realistically raw information mural as it is initially generated by the geometry mapper. The use of a non-photorealistic rendering style for the purpose of presenting the information objects possibly referring to data sets with the highest variation range of temperature and precipitation values is displayed in sub-figure 3(b). For the example presented, this is the day before the last.

Listing 2: Document excerpt for generating the presentation shown in Figure 3.

```
...
<sceneobjects>
  <object id="6" name="wideSpanningDay">
    <present method="1" start="PT11S" duration="PT10S" />
  </object>
</sceneobjects>
...
```

6 Conclusion

The presented work shows that the presentation of simulation results can be based on an XML script. This presentation description is based on the formal description of a temporal algebra for consecutive changes in a visualisation. The presented XML Schema provides a formalism to express syntactic, structural, and value constraints to any document instances and, therefore, to the description of temporally enhanced geometry presentation. An architecture of a flexible framework for implementing presentations by use of a script is introduced. This architecture is open to a wide range of potential application domains. An

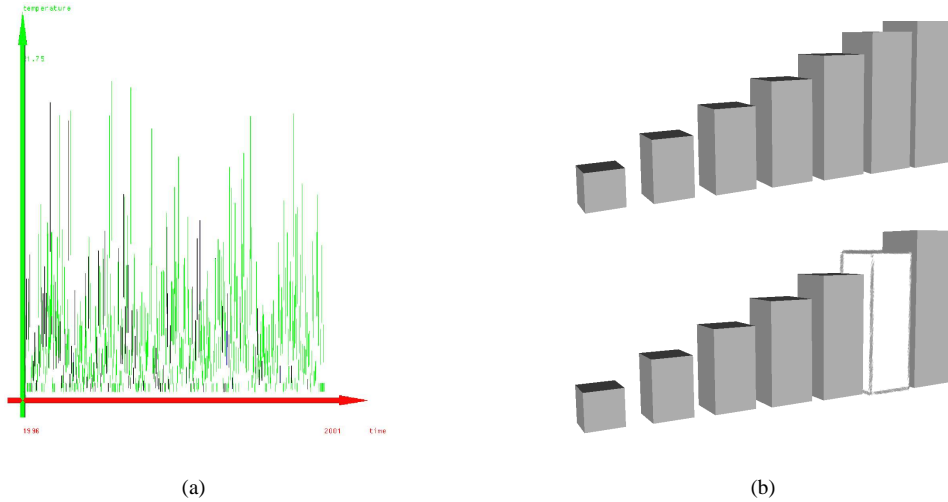


Figure 3: Enhancing an information mural presentation with NPR. The raw mural without special emphasis is shown to the left (a), (b) shows a zoomed in view on selected information objects. The upper images shows them in realistic shape whereas the bottom image outlines the use of NPR for the purpose of emphasising an information set according to user defined data parameters.

exemplary application shows the presentation of the result of a geometry mapping process typical for simulation data visualisation tasks.

The compression of XML for speedup is currently being investigated. In support of that, studies in slow network environments still have to be made. Future work includes—without being limited to—support for streaming documents and an integration of the script based presentation into a web service environment complying to the SOAP and WSDL standards.

Acknowledgements

This work is supported by the German Research Foundation (DFG) under grant number STR 398/4. Special thanks go to Roger Nelson of Washington State University (U.S.A.) for providing access to ClimGen as well as support in using it. Thanks to Axel Panning from the University of Magdeburg (Germany) who participated in the development of the presentation client presented in Section 5.2.

References

- James F. Allen. Time and Time again: The Many Ways to Represent Time. *International Journal of Intelligent Systems*, 6(4):341–355, 1991.
- Claudio Bettini, Sushil Jajodia, and Sean X. Wang. *Time Granularities*. Springer-Verlag, Berlin Heidelberg New York, 2000.
- Oliver Dunemann, Ingolf Geist, Roland Jesse, Kai-Uwe Sattler, and Andreas Stephanik. A Database-Supported Workbench for Information Fusion: Infuse. In Christian S. Jensen, Keith G. Jeffery, Jaroslav Pokorny, Simonas Šaltenis, Elisa Bertino, Klemens Böhm, and Matthias Jarke, editors, *8th International Conference on Extending Database Technology (EDBT) Proceedings*, Lecture Notes in Computer Science, pages 756–758. Springer, March 2002.
- Dean F. Jerding and John T. Stasko. The Information Mural: A Technique for Displaying and Navigating Large Information Spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):257–271, 1998.
- Roland Jesse. Motion enhanced Information Mural for Climate Data Visualisation. In Konrad Wojciechowski, editor, *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG 2002)*, volume 1, pages 374–379, 2002.
- Roland Jesse and Tobias Isenberg. Use of hybrid rendering styles for presentation. In *Proceedings of WSCG 2003*, 2003. to appear.
- Roland Jesse, Felix Ritter, and Thomas Strothotte. Bewegung als präemptive Präsentationsvariable in einem interaktiven System. In Thomas Schulze, Stefan Schlechtweg, and Volkmar Hinz, editors, *Simulation und Visualisierung 2002*, pages 275–288, Magdeburg, März 2002. Otto-von-Guericke Universität, SCS European Publishing House.
- Matthias Kreuseler and Heidrun Schumann. A Flexible Approach for Visual Data Mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):39–51, January-March 2002.
- Isabella Merlo, Elisa Bertino, Elena Ferrari, and Giovanna Guerrini. A Temporal Object-Oriented Data Model with Multiple Granularities. In C. Dixon and M. Fischer, editors, *Sixth International Workshop on Temporal Representation and Reasoning*, pages 73–81. IEEE Computer Society Press, 1999.
- C.W. Richardson and D.A. Wright. WGEN: A model for generating daily weather variables. Technical Report ARS-8, U.S. Department of Agriculture, Agricultural Research Service, 1984.