Representing Context-Dependent Information Using *Multidimensional XML*[*]

Yannis Stavrakas[1,2], Manolis Gergatsoulis[1], and Theodoros Mitakos[1]

[1] Institute of Informatics & Telecommunications,
National Centre for Scientific Research (N.C.S.R.) 'Demokritos',
153 10 Aghia Paraskevi Attikis, Greece.
[2] Knowledge & Database Systems Laboratory
National Technical University of Athens (NTUA), 157 73, Athens, Greece.
{ystavr,manolis}@iit.demokritos.gr
teo@softlab.ntua.gr

1 Motivation and Approach

XML (eXtensible Markup Language) is emerging as a new standard for data representation and exchange over the Web [3]. It is a markup language that resembles HTML, but unlike HTML, it focuses on the structure of data rather than on their presentation. The extensibility of XML makes it an ideal candidate for integration and manipulation of Web data through a common data model. A large number of DTDs (XML Document Type Definitions) that target all sorts of information domains has already been developed, and new DTDs are released at a fast pace. XML claims to be the enabling technology for application interoperability and for a unified view of heterogeneous information.

However, the nature of the Web poses a number of new problems. On the one hand, in traditional databases and information systems the number of users is more or less known and their background is to a great extend homogeneous. On the other hand, Web users do not share the same background and do not apply the same conventions when interpreting data. Such users can have different perspectives of the same entities, a situation that should be taken into account by Web data models. A similar problem appears when integrating information from various sources, where the same conceptual entity may exhibit different structure, or contain conflicting data.

Those problems call for a way to represent information entities that manifest different facets whose contents can vary in structure and value. Such cases are frequent in the frame of WWW where information providers cannot assume much about the needs and background of information consumers. As a simple example imagine a report that must be represented at various degrees of detail and in various languages. A solution would be to create a different XML document for every possible combination of variations. Such an approach is certainly not

[*] This work has been partially supported by the Greek General Secretariat of Research and Technology under the project "Executable Intensional Languages and Intelligent Applications in Multimedia, Hypermedia and Virtual Reality" of ΠENEΔ'99.

This paper appears in the Proceedings of the 4th International Conference "Research and Advanced Technologies for Digital Libraries" (ECDL' 2000), Lisbon, Portugal, Sept. 2000, J. Borbinha and T. Baker (editors), Lecture Notes in Computer Science LNCS 1923, pages 368-371, Springer Verlag, 2000.

practical, since it involves excessive duplication of information. What is more, different variants are not associated as being parts of the same entities. Therefore, although XML is in principle capable to represent such multifacet entities, it falls short for a number of reasons, namely (a) hidden semantics: XML does not address directly the issue of multiple facets, therefore it is the responsibility of an application to assign semantics in an ad-hoc manner, (b) cumbersome syntax: representing multiple facets of an entity in XML cannot be done in an elegant way, and (c) duplication of information: information that is common is duplicated, which is undesirable.

In this paper we describe an extension of XML, called *Multidimensional XML (MXML)* [6] that incorporates elements of Modal Logics [4] in XML in order to tackle the aforementioned problems. In an MXML document each information piece may be qualified by a *context specifier* denoting the *world* in which the information piece holds. There is a number of alternative worlds for a document, generated by the *dimensions* used in that document. The idea of applying dimensions to languages is not new [2, 5]. GLU [2] is a programming language based on the notion of multiple dimensions and ISE [7] is a multidimensional version of Perl. Our work was influenced by *Intensional HTML* (IHTML) [8], a Web authoring language that allows a single Web page to have different variations and to dynamically adapt itself to a given context.

2 MXML Syntax and Semantics

In an MXML document, any number of dimensions may be applied to elements and/or attributes to form *multidimensional elements* and *multidimensional attributes* whose content and value depend on their *context*. For a single multidimensional entity, contexts qualify the alternative facets of the entity. Note that, in MXML, conventional XML elements/attributes may coexist with multidimensional ones.

A multidimensional element has the form:

```
<@element_name>
   [context_specifier_1]
      <element_name attribute_specification_1>
          element_contents_1
      </element_name>
   [/]
       . . .
   [context_specifier_N]
      <element_name attribute_specification_N>
          element_contents_N
      </element_name>
   [/]
</@element_name>
```

where `element_contents_i`, with $1 \leq \mathtt{i} \leq N$ is the content of the (multidimensional) element for the context specified by [`context_specifier_i`]. The same element name is used for all *context elements*, while the *proxy element* groups context elements and is denoted by preceding the element name with the special symbol "`@`". A *context specifier* is a list of dimension assignments.

In case an attribute depends on dimensions the **`attribute_specification`** has the form:

```
attribute_name = [context_specifier_1] attribute_value_1 [/]  . . .
                 [context_specifier_n] attribute_value_n [/]
```

In the following example, the imaginary menu of a restaurant is represented as an MXML document.

```
<menu>
   <salad vegetarian = [season = summer] "yes" [/][default] "no" [/] >
      <name> Chef's salad </name>
      <@comment>
         [language = English, detail = low]
            <comment> A traditional salad. </comment>
         [/]
         [language = English, detail = high]
            <comment>
               A salad, with a long history which
               is connected with the tradition of the town.
            </comment>
         [/]
         [language = French, detail in {low, high}]
            <comment> Une salade regionale traditionelle. </comment>
         [/]
      </@comment>
      <@price>
         [season = summer] <price> 3 USD </price> [/]
         [default] <price> 4 USD </price> [/]
      </@price>
      <ingredient> tomato </ingredient>
      <@ingredient>
         [season != summer] <ingredient> bacon </ingredient> [/]
      </@ingredient>
         ... other salad ingredients ...
   </salad>
     ... other salads ...
</menu>
```

In this example, depending on the season of the year, the salad may contain different ingredients. Customers of the restaurant are mainly english-speaking and french-speaking. The meaning of **`[default]`** for a context element or attribute is that the context element or attribute holds for all the worlds for which no other context element/attribute of the same multidimentional entity holds.

For representing MXML documents we have extended *Object Exchange Model (OEM)* [1] to accommodate context-dependent information. Context specifiers of a multidimensional entity must be mutually exclusive. That is, for a single world no more than one context element/attribute may hold in a multidimensional element/attribute. An important issue concerns the inheritance of contexts throughout the paths of the document graph. In general, contexts are inherited and further specialized as we move down a path starting from the document graph root. Note that for any world - derived by assigning a specific value to each dimension - the MXML document is reduced to the conventional XML document which holds for that world.

3 Discussion and Future Work

In this paper we briefly describe a language for representing and exchanging context-dependent information over the Web. With its simple syntax and semantics, MXML addresses a common problem in WWW information representation, namely that of describing in a robust and concise way data which appear in different variations.

Our current research directions include: (a) The formalization of semantics, and the definition of a grammar for a Multidimensional DTD. The design and implementation of a query language, tailored to handling context-specific information. (b) The possible applications of MXML seem to have a lot of potential. Using MXML to encode GIS geographical information is in our immediate plans. The representation of time-dependent information using MXML is promising, since time can be seen as an MXML dimension. A very interesting direction is the use of MXML for integration of heterogeneous information.

References

1. S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann Publishers, 2000.
2. E. A. Ashcroft, A. A. Faustini, R. Jagannathan, and W. W. Wadge. *Multidimensional programming*. Oxford University Press, 1995.
3. T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible markup language (XML) 1.0. http://www.w3.org/TR/REC-xml, 1998.
4. Brian F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, 1980.
5. M. A. Orgun and W. Du. Multi-dimensional logic programming: Theoretical foundations. *Theoretical Computer Science*, 158(2):319–345, 1997.
6. Y. Stavrakas, M. Gergatsoulis, and P. Rondogiannis. Multidimensional XML. In *Distributed Communities on the Web, Third International Workshop (DCW'2000)*, Lecture Notes in Computer Science (LNCS) 1830, pages 100–109. Springer-Verlag, 2000.
7. P. Swoboda and W. W. Wadge. Vmake and Ise General Tools for the Intensionalization of Software Systems. In M. Gergatsoulis and P. Rondogiannis, editors, *Intensional Programming II*, pages 310–320. World Scientific, 2000.
8. W.W. Wadge, G. D. Brown, m.c. schraefel, and T. Yildirim. Intensional HTML. In *Proceedings of the Fourth International Workshop on Principles of Digital Document Processing (PODDP '98)*, Lecture Notes in Computer Science (LNCS) 1481, pages 128–139. Springer-Verlag, March 1998.