XML-based Integration of Hypermedia Design and Component-Based Techniques in the Production of Educational Applications

Antonio Navarro, José Luis Sierra, Baltasar Fernández-Manjón, Alfredo Fernández-Valmayor
Dpto. Sistemas Informáticos y Programación. Escuela Superior de Informática, Universidad Complutense de Madrid. Madrid – Spain: {anavarro, jlsierra, balta, alfredo } @sip.ucm.es

Keywords: Educational Hypermedia, XML, Document Transformations, Software Components, Design Methodology

Abstract: This paper describes a XML-based solution for developing educational hypermedias. This solution is the outcome of the lessons learned in the development of Galatea application, and integrates a hypermedia design methodology (ODH) with a generic technique for the construction of XML-based applications (DTC). Using ODH, designers can describe application's content, presentation and interaction as XML documents. DTC copes with the development of XML based applications by linking XML documents with component-based software, using document transformations as needed. The combination of both approaches leads to a solution for educational hypermedia development that overcomes some of the main problems that appear in the construction of these applications (communication between educators and developers, explicit representation of the educational strategy that underlies the application, development complexity and maintainability).

1. INTRODUCTION

The integration of new information and communication technologies enables the provision of a whole range of educational applications embodying new paradigms for learning [4]. Most of these new applications use hypermedia based design because it facilitates learner training through concept linking at the same time that provide students with compelling interactions and feedback [8].

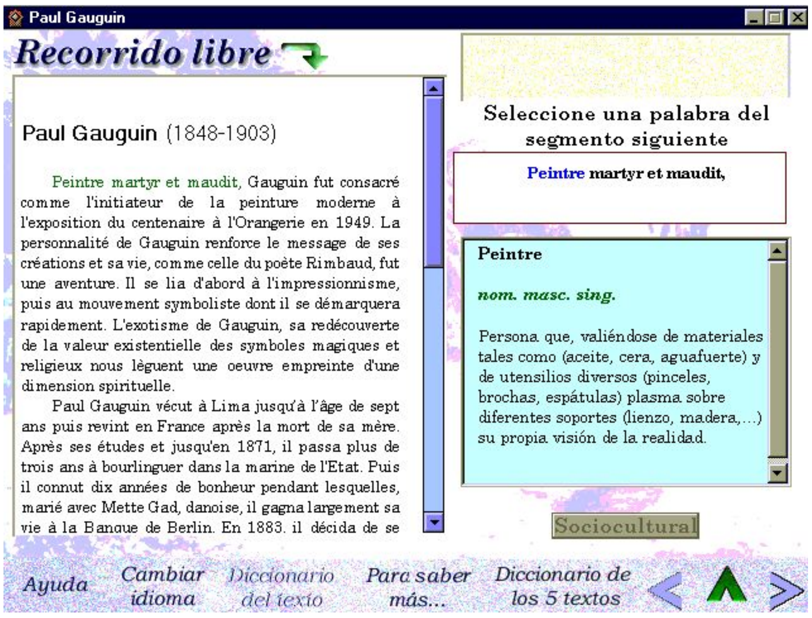
Despite of the advantages provided by this approach, the development of an educational hypermedia has specific requirements that are not easy to meet: (1) the need to base the application on solid educational criteria, (2) the need to facilitate the communication between customers (educators) and developers (software experts building the application), (3) the need to lower the development cost when requirements going beyond the scope of authoring environments (e.g. learner evaluation) and (4) the need to maintain an application, where a large amount of highly-structured information sources (e.g. text, exercises, learning plans) coexists and are potentially subjected to permanent revision by the educational team. We have identified all these drawbacks during the development of Galatea, a hypermedia educational project for foreign language text comprehension. From lessons learned during this project we are developing an specific hypermedia design methodology called ODH (Over-markup Design of Hypermedia applications), and a generic technique for the development of computer applications called DTC (structured Documents, document Transformations and software Components). Both approaches are based on the application of XML technologies, and although they are independent techniques, they can be combined to solve some of the problems of educational hypermedia development.

The structure of the paper is as follows. Section 2 describes Galatea and its educational strategy. Section 3 briefly describes the ODH methodology, and the solution given to the need of capturing the educational strategy making easier interaction between educators and developers. Section 4 summarises the DTC approach, and the solution given to the needs for software maintainability and reusability. Section 5 discusses the integration of both techniques and gives an example of this integration. Finally, section 6 shows some conclusions and future work.

2. GALATEA PROJECT

Included in SOCRATES/LINGUA EU effort, we have been working for three years in the Galatea project [2] aimed at developing a set of multimedia/hypermedia tutorials for the written and oral comprehension of the Romance languages (Fig. 1).

Galatea educational strategy is centred on the learning processes that arises in text reading comprehension of typologically related languages. This main strategy is implemented in the *Guided Route* module and it is complemented by a free text interaction (*Free Route*) plus a phonetic module to obtain a more natural understanding of the text. In this paper we will focus on the *Free Route* module that is conceived as a separate one where learner can freely interact with the same text that is considered in *Guided Route*. In this way the text is presented to the learner who can select any sentence of the text, and any word of these sentences to verify its contextual meaning according to some *contextual dictionary*.



Paul Gauguin (1848-1903)

Peintre martyr et maudit, Gauguin fut consacré comme l'initiateur de la peinture moderne à l'exposition du centenaire à l'Orangerie en 1949. La personnalité de Gauguin renforce le message de ses créations et sa vie, comme celle du poète Rimbaud, fut une aventure. Il se lia d'abord à l'impressionnisme, puis au mouvement symboliste dont il se démarquera rapidement. L'exotisme de Gauguin, sa redécouverte de la valeur existentielle des symboles magiques et religieux nous lèguent une oeuvre empreinte d'une dimension spirituelle.

Paul Gauguin vécut à Lima jusqu'à l'âge de sept ans puis revint en France après la mort de sa mère. Après ses études et jusqu'en 1871, il passa plus de trois ans à bourlinguer dans la marine de l'Etat. Puis il connut dix années de bonheur pendant lesquelles, marié avec Mette Gad, danoise, il gagna largement sa vie à la Banque de Berlin. En 1883, il décida de se

Seleccione una palabra del segmento siguiente

Peintre martyr et maudit,

Peintre

nom. masc. sing.

Persona que, valiéndose de materiales tales como (aceite, cera, aguafuerte) y de utensilios diversos (pinceles, brochas, espátulas) plasma sobre diferentes soportes (lienzo, madera,…) su propia visión de la realidad.

Figure 1. Galatea screen snapshot

3. ODH APPROACH

ODH [7] is a hypermedia design methodology that combines the eXtensible Markup Language XML [13] with object-oriented software development techniques [1,9,10]. ODH uses XML techniques to capture the educational strategy and to alleviate the communication problems between educators and developers. In addition, ODH uses object oriented techniques to describe those processing activities that need to be included in the final application (e.g. exercise evaluation).

Educational strategy is captured by ODH using an XML DTD (*Document Type Definition*) called the *content DTD*. This construction describes the structure of the application's contents and the hyperlinks between these contents. Actual contents are directly provided by educators in a structured *content document* conforming the content DTD. Because content DTD defines a language close to educators' perceptions, the educators' team understand the structure of the document and they are able to write it using a XML editing tool. On the other hand, these XML documents can be automatically processed by final applications. In this way, use of the content DTD improves the interaction between educators and programmers, and therefore the maintenance of the application, because any change made by educators in the contents can be reflected in the final application without coding. Fig. 2 illustrates a fragment of the Galatea content document corresponding to the contextual dictionary.

```
..................
<text>
 <title>Paul Gauguin 1848-1903</title>
     <paragraph>
          <sentence refToSpliSent="ss1">Peintre martyr et maudit
                                                   </sentence>
          <sentence refToSplitSent="ss2">,Gauguin fut consacre comme
l'initiateur de la peinture moderne a l'exposition du centenaire a
l'Orangerie en 1949.</sentence>
..................
```

Figure 2. A fragment of Galatea content document

ODH uses another XML DTD, called the *presentation DTD* to characterize the presentational structure of hypermedia applications. The elements of the presentation DTD describe the application presentational GUI elements (screens, windows, buttons, etc.) and the hyperlinks between them.

```
..................
<!ELEMENT screen (header, links?,(window|button)+)>
<!ATTLIST screen name ID #REQUIRED
                 initial (yes|no) "no "
                 %presAtt1;>
<!ELEMENT header (#PCDATA)>
<!ATTLIST header %presAtt2;>
<!ELEMENT window (header?,element,diagrams?,help?)>
<!ATTLIST window name NMTOKEN #REQUIRED
                  type (frame|picture|virtual) "frame"
                  %presAtt1;>
<!ELEMENT element (content, pres?)>
..................
```

Figure 3. A fragment of ODH presentation DTD

The instance of the presentation DTD is called *design document*. The separation of content and presentation provides the means to associate different presentations to the same information, allowing to reuse the same content in different views that can change in order to accommodate to learner skills. Fig. 3 illustrates part of the presentation DTD.

The relationship between content DTD and presentation DTD is accomplished through *over-markup*. The basic over-markup idea is that the design document is built in such a way that the contents of the elements of the presentation DTD are chunks of the instance of the content DTD (the content document). That is, during over-markup the contents of the hypermedia application are bound to their presentational layer, to represent the whole hypermedia application.

```
<presentation  >
  <screen  name="contDic" initial=" no" xPos="0" yPos="0" length="640"
wide="480">
  <header  font="garamond" size="18" bold="noB" italic="noI"/>
  <window  name="cdWindow1" type="picture" xPos="0" yPos="0" length="125"
wide="200">
    <element >
      <content >
      <text>
       <title>Paul Gauguin 1848-1903</title>
  <sentence refToSplitSent="ss1">Peintre martyr et maudit,
</sentence>
        <sentence refToSplitSent="ss2">Gauguin fut consacre comme
l'initiateur de la peinture moderne a l'exposition du centenaire a
l'Orangerie en 1949.</sentence>
..........................................
      </content >
      <pres ><presAtt  font="garamond" size="12" bold="noB"
italic="noI"/></pres >
     </element >
..........................................
  </window >

..........................................
 </screen >
</presentation  >
```

Figure 4. Simplified overmarkup example

In addition to presentational aspects, the design document is also involved with other computational activities. These additional complex computational activities (e.g. an exercise that evaluates the learner knowledge) are represented using object-oriented diagrams (mainly class and state transition diagrams) that are attached with the design document. In this way, this document provides with a real representation of the total application used by educators and programmers. Educators use the design document to evaluate if it conforms its requirements, and make any change (obviously they ignore the object-oriented diagrams). Developers use part of this document in the coding phase directly, whereas other parts represents the application design that they must translate in real code. The content DTD and the presentation DTD related through over-markup provide a common framework to educators and developers improving their communication. They also help to capture the educational strategy that underlies the application [2]. Fig. 4 illustrates an over-markup example where design document shows the content of the text for the contextual dictionary appearing in a window corresponding to the screen of the contextual dictionary (elements of presentation DTD appear in boldface, and elements of content DTD appear in italic).

4. DTC APPROACH

DTC [11] enables to develop XML-based applications combining XML documents, (reusable) software components [12], and document transformations [6].

For building an application according DTC, first of all, the different kinds of information to be used in the application must be provided as a collection of XML documents. These documents are jointly named as *application documents* and they can be classified into two main categories: (a) *domain documents*, that are those documents containing domain specific information that could be reused across different applications (e.g. a dictionary) and (b) *application dependent documents* that are documents that only have meaning inside a single application (e.g. presentational information for a map, or a GUI layout description).

In parallel to the application document provision, the *application software* is built using software components. Each DTC component is able to process a kind of *componential documents* giving them an operational meaning. In addition component interfaces are also described in XML terms. DTC components can be classified in several categories. *Primitive components* are the basic building blocks for the application construction. *Containers* allow the aggregation of component conglomerates. Primitive components are subdivided into *markup interpreters* (devoted to give operational support for abstract markup languages), *primitive facilities* (components that carry out some basic functionality in the final application) and *transformers* (components for adapting information flows between other components). Containers are divided into *GUI containers* (for displaying visual representations of their sub-components) and *controllers* (for describing the behaviour of their sub-components).

The last step is to integrate all the application ingredients (documents and components) in the final application. Many times, both domain documentation and components are reused, so that multiple disagreements in information structure (between application and componential documents, and between component interfaces in the application software) can arise. Given that all the information accepted by DTC components is XML structured, document transformations can be used as a unified mechanism for solving all these disagreements [5, 6, 14]. Transformations are effectively incorporated into DTC using transformer components. Transformers are used for building *transformations engines* that are executed off-line and enable to obtain componential documents for application ones. They also are used on-line as information-flow adapters between other components.

DTC allows to reuse pre-existing documents, because with transformations it is possible their integration in the executable application. In addition, DTC eases to reuse software components, using transformations as a convenient glue. Because domain information can be written in domain terms instead of application software terms, DTC also encourages application maintainability. Domain experts can be engaged on maintaining that information while application builders can make the required tunings in the application-specific documentation in order to incorporate the changes, many times with zero programming effort.

5. ODH AND DTC INTEGRATION

This section outlines how ODH and DTC can be naturally integrated in a unified solution for the development of educational hypermedia, and illustrates it with a simple example concerning Galatea application.

5.1 INTEGRATING THE APPROACHES

ODH and DTC are independent approaches, but, because both of them are based on XML technologies, they can be easily integrated in the development of complex educational hypermedia. This integration is schematised in Fig. 5.

In this integrated framework, hypermedias are designed with ODH methodology obtaining the content and design documents given in XML terms. Then DTC approach is used for implementing the final application.

DTC takes as input ODH content and design documents. Because constructs in an ODH presentation DTD have clear intended presentational and/or interaction semantics, initial application software is devised in terms of such semantics. Other behaviours, given in ODH as a collection of object oriented diagrams, are used to complete the assemblage of the application software. Additional information for deriving the final application is introduced as DTC application dependent documents. Derivation of DTC componential documents is driven by the ODH design document. The DTC off-line transformational process uses this document for indexing the overmarked content fragments. Then it transforms these contents as needed. In addition, presentational metainformation is used for setting presentational features of the GUI components in the application software, using additional application-dependent information as needed.

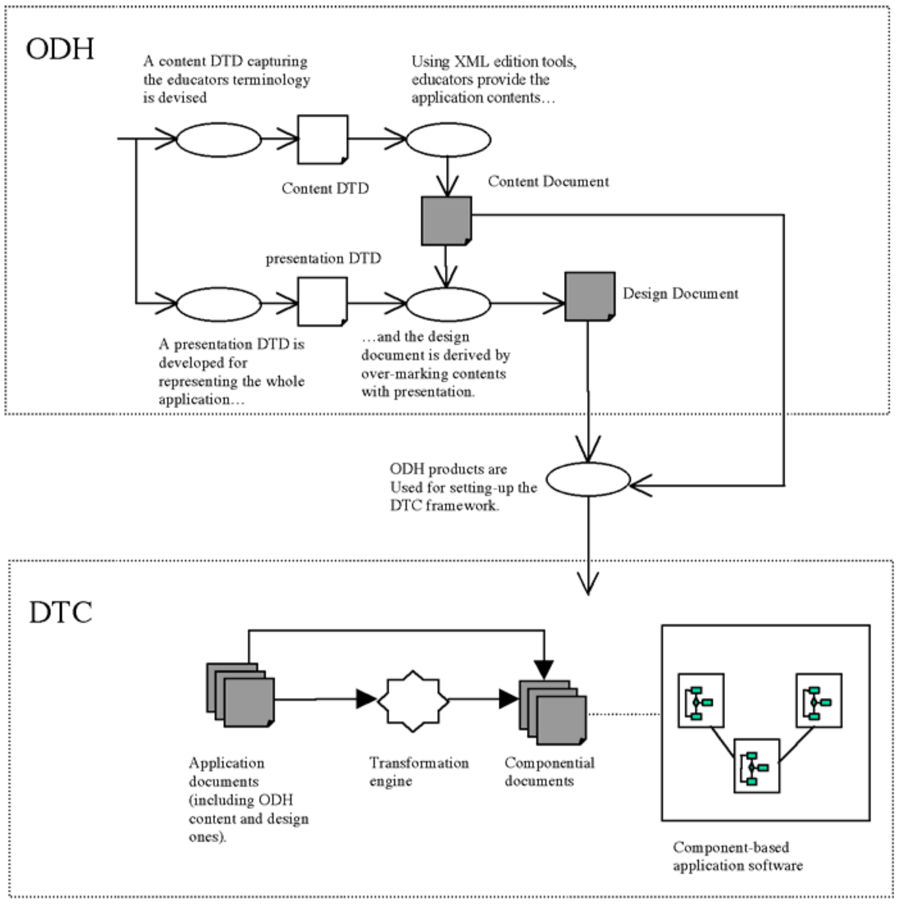
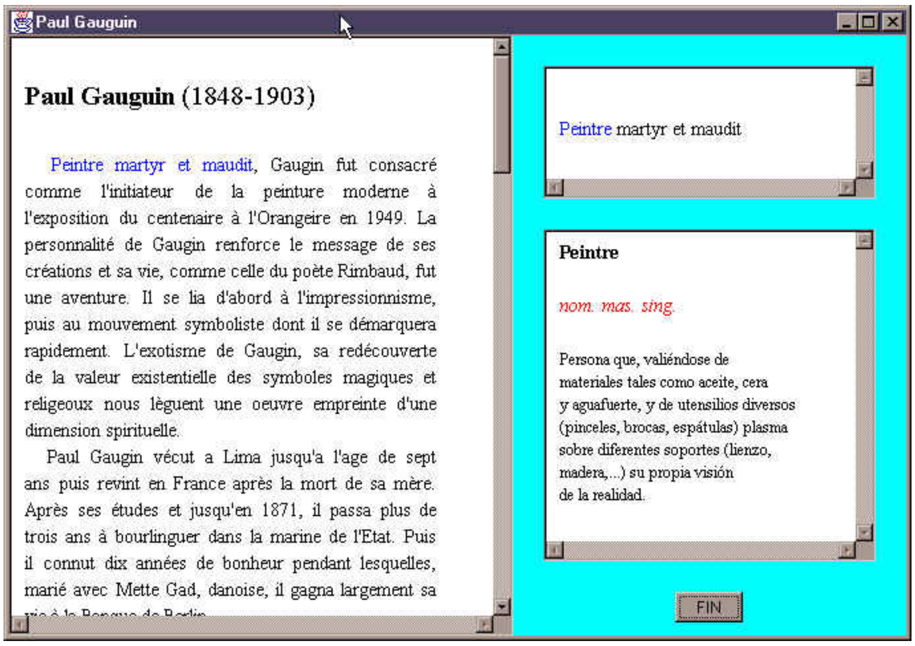

ODH
A content DTD capturing the educators terminology is devised
Using XML edition tools, educators provide the application contents…
Content DTD
Content Document
presentation DTD
Design Document
A presentation DTD is developed for representing the whole application…
…and the design document is derived by over-marking contents with presentation.
ODH products are Used for setting-up the DTC framework.
DTC
Application documents (including ODH content and design ones).
Transformation engine
Componential documents
Component-based application software

Figure 5. Integration of the ODH and DTC approaches

5.2 AN EXAMPLE

This example uses a simplified version of the Galatea *Free Route* module to illustrate the integration of ODH and DTC. The main content associated with this module is the contextual dictionary. Educators (i.e. the linguistic team) use an XML editor for producing ODH contents such as that shown in Fig. 2. These contents are over-marked by developers in terms of the presentation DTD (Fig. 3) to configure the final appearance of the module. The resulting design information is encoded in XML terms (as shown in Fig. 4).



Paul Gauguin (1848-1903)

Peintre martyr et maudit, Gaugin fut consacré comme l'initiateur de la peinture moderne à l'exposition du centenaire à l'Orangeire en 1949. La personnalité de Gaugin renforce le message de ses créations et sa vie, comme celle du poète Rimbaud, fut une aventure. Il se lia d'abord à l'impressionnisme, puis au mouvement symboliste dont il se démarquera rapidement. L'exotisme de Gaugin, sa redécouverte de la valeur existentielle des symboles magiques et religeoux nous lèguent une oeuvre empreinte d'une dimension spirituelle.

Paul Gaugin vécut a Lima jusqu'a l'age de sept ans puis revint en France après la mort de sa mère. Après ses études et jusqu'en 1871, il passa plus de trois ans à bourlinguer dans la marine de l'Etat. Puis il connut dix années de bonheur pendant lesquelles, marié avec Mette Gad, danoise, il gagna largement sa

Peintre martyr et maudit

Peintre

nom. mas. sing.

Persona que, valiéndose de materiales tales como aceite, cera y aguafuerte, y de utensilios diversos (pinceles, brocas, espátulas) plasma sobre diferentes soportes (lienzo, madera,...) su propia visión de la realidad.

Figure 6. Screen snapshot of the (simplified) application produced by the integration of ODH and DTC

Once these ODH content and design documents are available, DTC can be applied for building this module. According the intended semantics attached with the presentation elements, the DTC application software can be built. This software mainly relies on a `SelectableText` markup interpreter that enables the selection of text fragments, and a `Navigator` control component enabling the implementation of the hypermedia relationships for the module. Once the application software has been devised, a transformation engine is configured for producing the needed componential documents. These documents include, with other minor stuff, text-level and sentence-level contents for occurrences of `SelectableText` and a description of the navigational control document for the `Navigator` control component. Fig. 6 shows the resulting application. DTC components used in this example has been implemented using Java together with the Oracle XML Parser [3] and its associated XSLT (eXtensible Stylesheet Language Transformations [14]) support.

6. CONCLUSIONS AND FUTURE WORK

The development of educational hypermedias is a costly task that involves several experts with very different backgrounds. The ODH design methodology improves the communication between educators and developers, and provides the means to capture the educational strategy that underlies the application. On the other hand, the DTC approach to the construction of XML-based applications eases application maintainability and different levels of reuse. DTC goals are achieved by a coherent integration componentware and markup technologies in a unified framework.

The integration of ODH and DTC allows a more rational development of (educational) hypermedia. Design document provides a complete guide to educators and developers for building the application. In this way, if a change would be needed (e.g. in the contextual dictionary) educators can edit directly the content document, and automatically the DTC framework rebuilds the new application with zero programming effort.

Next steps in the project are the development of CASE tools for supporting the application of the ODH and DTC processes in a sound and user-friendly way, and the definition of an object-oriented semantics for ODH presentation DTD elements. In addition we are considering the definition of DTC extensions for allowing the derivation of domain specific document editors.

ACKNOWLEDGEMENTS

The EU project Galatea (TM-LD-1995-1-FR89) and the Spanish Committee of Science and Technology (TIC97 2009-CE and TIC98-0733) have supported this work.

7. REFERENCES

1. Booch G., *Object-oriented analysis and design with applications*, Second Edition, Benjamin Cummings Publishing Company, 1994.
2. Fernandez-Valmayor A., Lopez-Alonso C., Sere A., Fernandez-Manjon B., A hypermedia design for learning foreign language text comprehension, IFIP WG3.2/WG3.6 August 1999 *Working Conference on Building University Electronic Educational Environments*,

Proceedings, Co-editors Stephen D. Franklin and Ellen Strenski. Kluwer Academic Publishers, in press.

3. http://technet.oracle.com/
4. Ibrahim, B., Franklin, S.D., 1995. “Advanced Educational Uses of the World Wide Web”. *Computer Networks and ISDN Systems*, vol 27, no 6, pp. 871-877.
5. International Standards Organization. “Document Style Semantics and Specification Language (DSSSL)”. ISO/IEC 10179. 1996.
6. Kuikka, E. Pentonnen, M. “Transformation of Structured Documents”. Tech. Report CS-95-46. University of Waterloo. 1995
7. Navarro, A. Fernández-Manjón, B. Fernández-Valmayor,A. J.L.Sierra. “A Practical Methodology for the Development of Educational Hypermedias”. Accepted in *ICEUT2000. IFIP-WCC2000*.
8. Norman D.A., Spohrer J.C., Learner-Centred Education, *CACM* 39 (4) 24-27, 1996.
9. Rumbaugh J., Blaha M., Premerlani W., Eddy F., Lorenzen W., *Object-Oriented Modeling and Design*. Prentice Hall, 1991.
10. Rumbaugh J., Booch G., Jacobson I., *Unified Modeling Language Reference Manual*, Addison-Wesley Object Oriented Series, 1998.
11. Sierra, J.L. Fernández-Manjón, B. Fernández-Valmayor,A.Navarro, A. “Integration of Markup Languages, Document Transformations and Software Components in the Development of Applications: the DTC Approach”. Accepted in *ICS2000. IFIP-WCC2000*.
12. Szyperski, C. *Component Software: beyond Object-Oriented Programming*. Adisson Wesley. 1998.
13. W3C, Extensible Markup Language XML Version 1.0, http://www.w3.org/TR/REC-xml, 1998.
14. W3C, XSL Transformations (XSLT) Version 1.0, http://www.w3.org/TR/xslt, 1999.