

AUTHORING MULTIMEDIA DOCUMENTS THROUGH GRAMMATICAL SPECIFICATIONS

Jun KONG

Meikang QIU

Kang ZHANG

Department of Computer Science, P.O. Box 830688, EC31, The University of Texas at Dallas,
Richardson, TX 75080-0688, USA
kzhang@utdallas.edu

ABSTRACT

The diversity of multimedia devices is an important feature on the Internet, which demands an executable mechanism to intelligently adjust the appearance of a document according to different viewing contexts. In order to satisfy this requirement, a graph-grammar-based approach for multimedia authoring is presented by extending a context-sensitive graph grammar formalism, *Reserved Graph Grammar (RGG)*, with the capability of spatial specifications. Based on a parser for the RGG, a graph grammar functions as a mapping from the structure of a document to a desirable layout, and graph transformations are automatically performed when contexts are changed.

1. INTRODUCTION

Multimedia authoring is a challenging issue when considering the diversity and various capabilities of media devices. An authoring tool should provide an easy approach to not only specifying the structural relations with spatial and temporal relations among medial objects, but also meeting the dynamic requirements, i.e. the authoring tool should provide a mechanism to support the intelligent adaptation of appearances according to different contexts. Fortunately, the graph grammar is a possible solution.

Motivated by the theoretical support for pattern recognition, compiler construction, and data type description, graph grammars originated in the late 60 have provided a theoretically sound and well-established foundation [Roz97]. Graph grammars are powerful in defining logic relations among constructs and providing a natural approach to specifying the hyper-graph structure of a multimedia document. Especially, the *Reserved Graph Grammar (RGG)* is a context-sensitive graph grammar formalism with a parsing complexity of polynomial time under a non-ambiguous condition [Zha01]. The applications of the RGG [Zha01a] [Kon03] demonstrate its powerful expressiveness.

However, we cannot apply the RGG directly to multimedia authoring, since it is unable to specify how

constructs look like, one of the important aspects in multimedia authoring. This paper presents a spatial extension to the RGG and illustrates its application in multimedia authoring.

The rest of the paper is organized as the following. Section 2 illustrates how to extend the RGG with spatial specifications. Section 3 presents our approach to multimedia authoring and goes through an example. Section 4 lists related works, followed by the conclusion in Section 5.

2. EXTENDING RESERVED GRAPH GRAMMAR

2.1. Graph grammar notations

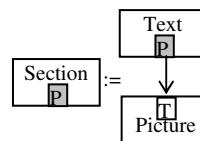


Figure 1 A production in RGG

The RGG is a context-sensitive graph grammar formalism with a parsing complexity of polynomial time under a non-ambiguous condition. In the RGG, nodes are organized into a two-level hierarchy as illustrated in Figure 1. A large rectangle is

the first level called a *super vertex* with embedded small rectangles as the second level called *vertices*. Each vertex is labeled by a unique character. A grammar rule (Figure 1) is called a *production* consisting of two sub-graphs, called *left graph* and *right graph*. A sub-graph in the host graph matching the left or right graph is called a *redex*. The RGG employs a marking technique to solve the embedding problem and avoid ambiguity. If a super vertex or a vertex is marked, it will reserve its outgoing edges connected to vertices outside the replaced sub-graph in the application of a production. A marked vertex is represented with a gray background (to save space, we use a circle with a unique integer to represent a marked vertex in the later example). The RGG is equipped with a deterministic parsing algorithm, called *selection-free parsing algorithm (SFPA)* trying only one parsing path. Zhang *et al.* proved that the time complexity of SFPA is polynomial under a non-ambiguous condition [Zha01].

The RGG provides a powerful mechanism to represent structures graphically and perform automatic syntax validation through an automatically generated

parser. In order to apply the RGG to the multimedia authoring, we enhance it with supporting spatial specifications.

2.2. Spatial specifications

Spatial knowledge consists of information about topology, direction, distance and shape etc. We are most interested in specifying direction and topological relations qualitatively.

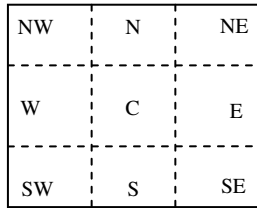


Figure 2 Direction specification

We divide a node's super vertex into nine areas as shown in Figure 2. The central area represents the super vertex itself. Surrounding the central area (C), the eight areas represent eight directions: N (North), S (South), E (East), W (West), NW (Northwest), NE (Northeast), SW (Southwest), and SE (Southeast). Each of these areas indicates the relative direction of the other node connected to the current node. Specifically, more than one node can be located in the same direction relative to the current node.

Specifically, more than one node can be located in the same direction relative to the current node.

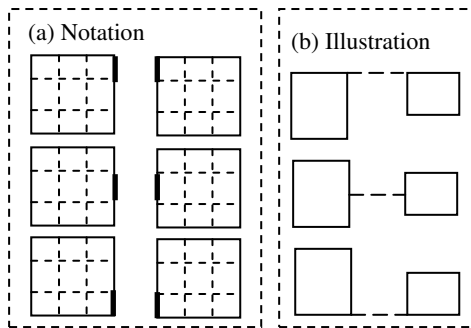


Figure 3 Alignment relations in horizontal direction

Two nodes can be aligned vertically or horizontally. In the vertical direction, we distinguish three sub-cases, i.e. the top, bottom or central lines of two nodes are aligned. Three sub-cases in horizontal direction are similarly defined. The boundary of a node is divided into 12 segments. A pair of bold segments is used to demonstrate the relation for alignment as illustrated in Figure 3.

We are interested in four types of topological relations: un-touching, touching, partially overlapping and containing as shown in Figure 4. Two nodes with solid boundaries represent the un-touching relation. Dotted lines on the boundaries of both nodes indicate the touched parts. Two nodes are partially overlapped if only one of the two nodes' boundaries is partially dotted. Specifically, if one node's boundary is totally dotted, one node is contained in the other node.

We qualitatively classify the distance into *close*, *remote* and *spring* relations, which are represented by different gray degrees. The three classes are not sufficient

to distinguish the distance among media objects. We attach to each node some attributes expressing absolute values or percentages of the distance in relative to other nodes.

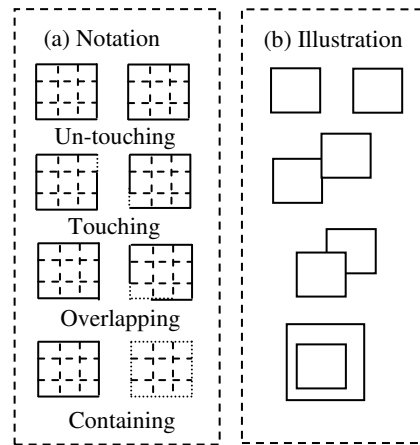


Figure 4 Topological relations

2.3. Event driven

In order to address the dynamic issue, we classify graph productions into conditional and unconditional ones. An application of an unconditional production is performed once its corresponding redex is matched in the host graph. A conditional production can only be applied when a specific event occurs, such as the change of the device capability, user's interaction etc. Since graph transformations can be performed under predefined conditions, the appearance of a multimedia document can be adjusted accordingly when the context is dramatically changed.

In summary, a production in the extended RGG mainly contains three aspects: structural, spatial and triggering-conditions specifications.

3. GRAPH-GRAMMAR-BASED APPROACH

The graph-grammar-based approach consists of four modules as shown in Figure 5: event authoring, event listener, production authoring and parser. Event authoring module lets the user describe the events, by which a transformation is activated. Event listener will monitor the system to check if any specific event occurs. Upon the user's inputs and messages dispatched by the event listener, the parser will perform corresponding graph transformations.

The production authoring module provides a tool to design a graph grammar according to user's description. The grammar not only illustrates how to construct a multimedia document through various media objects, but also how those constructs look like.

The parser validates the structure of a host graph, and a parsing tree, which reflects the hierarchical structure of a

document, is generated. A desirable layout is achieved through the visualization of spatial specifications based on the parsing tree. When the user modifies the font sizes, or device capability is changed etc, a message will be dispatched to the parser, and a conditional production can be triggered to perform a graph transformation. The layout, therefore, is adjusted to satisfy the new context. Specifically, the position of a composite node made up of other nodes is processed as one entity in later applications.

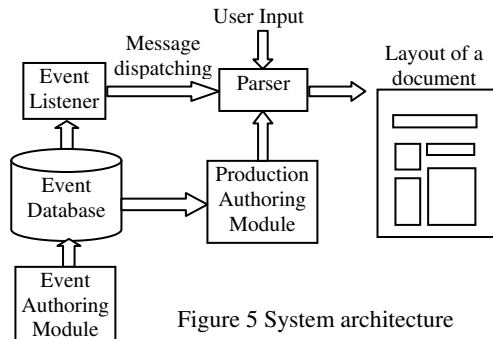


Figure 5 System architecture

With current document markup languages, the layout of a page is relatively static and fixed [Bor00]. When the user's requirement or the device capability is changed, the layout may become unsatisfied. The reason is that such markup languages do not provide sufficient information when the context is changed. With the graph-grammar-based approach, the appearance is adjusted dynamically to suit different environments.

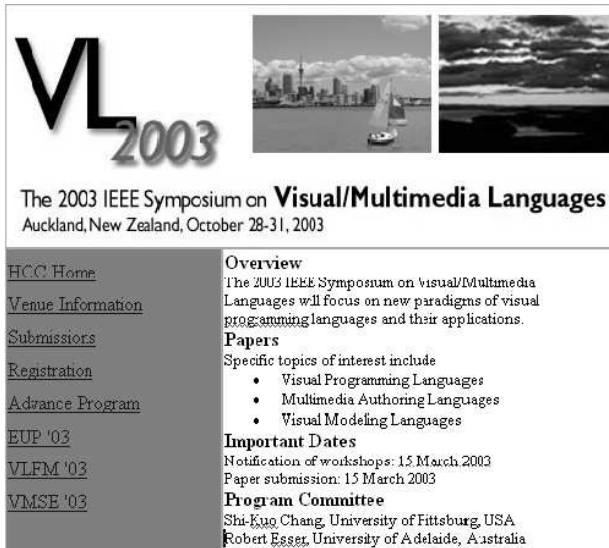


Figure 6 A multimedia document

With the help of graph grammars, we specify the spatial properties among constructs rather than assigning coordinates to constructs. When the properties of one object are changed, the properties of other objects can be adjusted accordingly. A group of conditional productions can represent one display style. Therefore, a document can have several groups of productions representing different styles, just like Cascading Style Sheets in HTML. When

the user is interacting with the system, such as clicking on a media object, a specific production is activated to indicate the move of user's focus. In addition to the dynamics, the approach also supports the syntax-directed authoring.

In order to illustrate our approach, we give an example using a conference website in Figure 6, and present its corresponding grammar in Figure 7. Production 2 indicates that such a document consists of four objects, a picture, a title, several links and contents. Moreover, it shows how those constructs look like. For example, the picture should be aligned with the title, and they are touched. Productions 3 and 4 specify how to combine several links together. Productions 6 and 7 show that contents are made up of several sections, each of which is constructed by a sub-title and relevant contents as illustrated in Production 5.

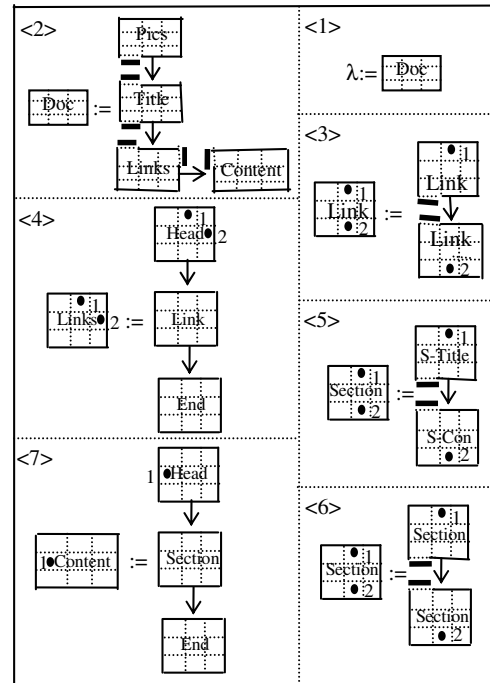


Figure 7 Graph grammar

4. RELATED WORK

There has been a number of systems and approaches for the presentation and design of multimedia systems, such as Comet [Fei93] and WIP [And93]. These systems employ some forms of rule-based mechanisms to represent graphical design knowledge. The rules control the search of all possible solutions and determine an appropriate solution. One of the most difficult issues in these systems is how to specify the control mechanism, which could be more easily addressed by a parsing algorithm for graph grammars [Wei94].

Most related to our work is that of Weitzman and Wittenburg [Wei94]. Weitzman and Wittenburg applied a graph grammar formalism – Relation Grammar, to the

automatic presentation of multimedia documents. The grammar governs the structure of a document. One or more parsing trees, each of which represents an independent presentation, are derived through a parser. Then, a syntax-directed translation is made on the tree. The final layout is created by a constraint solver following the translation. In this approach, relational grammar functions as a mapping from a representation of one style of multimedia documents to the forms that specify how to realize the media objects.

Inspired by the work of Weitzman and Wittenburg, Cruz and Lucas developed a visual querying and presentation system called Delannay^{MM} [Cru97], but grammars are not used in this system. Zhang *et al.* presented a visual approach to XML document design and transformation [Zha01a]. Rather than using DTD and XSL, the RGG is used to define the XML syntax and specify the transformations to other languages.

Constraints are very popular in user interface and interactive systems. Marriott *et al.* [Mar02] extends Scalable Vector Graphics format SVG with constraint-based specification. Such an extension supports client-side adaptation of documents to different viewing conditions. Borning *et al.* [Bor00] gave a constraint-based system for the layout of a Web document. In the system, both the author and the viewer can impose constraints on documents. Therefore, the final appearance of a document is determined by the negotiation between author and viewer. Also, Borning *et al.* imposed constraints in hierarchy to differentiate them as required or preferential ones. Compared to constraint-based systems, we are more interested in providing an approach to specifying the structural relations together with spatial relations. Furthermore, the key module in our system is a parser rather than a constraint solver.

5. CONCLUSION

The diversity of multimedia devices is an important feature on the Internet, which demands an executable mechanism to intelligently adjust the appearance of a document according to different viewing contexts. This paper presents an authoring approach based on the RGG with a spatial extension. A graph grammar functions as a mapping from a presentation of a style to a physical layout. The syntactical definition in the grammar captures the structure of documents and the parser performs automatic validation on the document. Rather than assigning every object with an absolute co-ordinate value, graph grammars specify how the document looks like by defining spatial relations in the grammar. By visualizing those relations, a layout is automatically generated.

The spatial relations expressed in RGG are not complete. For example, we have not visualized the relation that the bottom line of one object is aligned to the top line of the other object.

In addition to spatial specifications for media objects, temporal specifications offer another important dimension in the design of multimedia documents. Allen presented some common temporal relations such as *during*, *before*, *meet* relations [All83], which are potentially adaptable to the automatic presentation. Temporal specifications determine the sequence of presentation, so the ability of visually specifying them together with spatial and syntactic relations would be desirable for multimedia designers.

REFERENCES

- [All83] F. Allen, "Maintaining Knowledge about Temporal Intervals", *Communications of the ACM*, (26)11, pp. 832-843, 1983.
- [And93] E. Andre, W. Finkler, and W. Graf etc, "WIP: The Automatic Synthesis of Multimodal Presentations", *Intelligent Multimedia Interface*, M. MayBury (Ed.), AAAI Press/MIT Press, Cambridge, MA, pp. 75-93, 1993.
- [Bor00] A. Borning, R. K. Lin, and K. Marriott, "Constraint-based Document Layout For The Web", *Multimedia System*, 8, pp. 177-189, 2000.
- [Cru97] I. F. Cruz and W. T. Lucas, "A Visual Approach to Multimedia Querying and Presentation", *Proc. ACM Multimedia'97*, Seattle, USA, 8-14 Nov, 1997.
- [Fei93] S. Feiner and K. McKeown, "Automating the Generation of Coordinated Multimedia Explanations", *Intelligent Multimedia Interface*, M. MayBury Ed., AAAI Press/MIT Press, Cambridge, MA, pp. 117-138, 1993.
- [Kon03] J. Kong, K. Zhang, and M. L. Huang, "Application-oriented Spatial Graph Grammars", *Proc. IASTED on Applied Informatics (AI 2003)*, Innsbruck, Austria, Feb. 2003.
- [Mar02] K. Marriott, B. Meyer, and L. Tardif, "Fast and Efficient Client-side Adaptivity for SVG", *Proc. WWW'2002*, pp. 496-507, 2002.
- [Roz97] G. Rozenberg (Ed.), *Handbook on Graph Grammars and Computing by Graph Transformation: Foundations*, Vol.1, World Scientific, 1997.
- [Wei94] L. Weitzman and K. Wittenburg, "Automatic Presentation of Multimedia Documents using Relational Grammars", *Proc. ACM Multimedia'94*, San Francisco, USA, 15-20 Oct, 1994.
- [Zha01] D. Q. Zhang, K. Zhang, and J. Cao, "A Context-Sensitive Graph Grammar Formalism for the Specification of Visual Languages", *Computer Journal*, (44)3, pp. 187-200, 2001.
- [Zha01a] K. Zhang, D. Q. Zhang, and Y. Deng, "Graphical Transformation of Multimedia XML Documents", *Annals of Software Engineering*, 12, pp. 119-137, 2001.