

Formal specification and prototyping of multimedia applications

Timothy Arndt¹, Shi-Kuo Chang², Angela Guercio³

¹ Department of Computer and Information Science, Cleveland State University, Cleveland, Ohio, USA; arndt@cis.csuohio.edu

² Department of Computer Science, University of Pittsburgh, Pittsburgh, Pennsylvania, USA; chang@cs.pitt.edu

³ Dipartimento di Matematica ed Informatica, University of Salerno, Baronissi, Italy; ma@udsab.dia.unisa.it

Abstract. Multimedia systems incorporating hyperlinks and user interaction can be prototyped using TAOML, an extension of HTML. TAOML is used to define a Teleaction Object (TAO) which is a multimedia object with associated hypergraph structure and knowledge structure. The user can create and modify the private knowledge of a TAO so that the TAO will react automatically to certain events. The hypergraph structure supports the effective presentation and efficient communication of multimedia information. TAOs are valuable since they greatly improve the selective access and presentation of relevant multimedia information. In this paper, a formal specification methodology for TAOs using SR grammars is described. An attributed SR grammar is then introduced in order to associate knowledge with the TAO. The limitations necessary to achieve an efficient parser are given. The grammatical formalism allows for validation and verification of the system specification. This methodology thus provides a principled approach to specify, verify, validate and prototype multimedia applications.

Key words: Formal specification – Rapid prototyping – Multidimensional grammars - Distributed multimedia

1. Introduction

Recent developments in computer technology have enabled large, distributed multimedia applications to be created in such application areas as education [25], health care [24], and process control [15]. These applications are often web-based and involve a large amount of user interaction. All of these characteristics increase the complexity of designing, coding and testing. The prototyping of multimedia applications based upon software engineering principles has not yet been adequately addressed by the research community although recently research interest in the area of multimedia and software engineering has increased. An indication of this increased interest is the convening of the first International Workshop on Multimedia Software Engineering held in April 1998 as part of the International Conference on Software Engineering [16]. In this paper, we apply software engineering methodology to the production of multimedia applications introducing a principled approach to specify, verify, validate and prototype such applications.

Our approach to multimedia application development is based on a collection of tools which support the creation of Teleaction Objects (TAOs) [5,14]. A TAO is a multimedia object with associated hypergraph structure and knowledge structure. The user can create and modify the private knowledge of a TAO so that the TAO will react automatically to certain events. The knowledge structure of a TAO is an active index (IX) [6] which consists of a collection of index cells (ICs). The hypergraph structure supports the effective presentation and efficient communication of multimedia information. The static aspects of the hypergraph structure are described by a Multimedia Static Specification (MSS). TAOs are valuable since they greatly improve the selective access and presentation of relevant multimedia information. The tools described in this

paper provide a way to formally specify the TAOs comprising the application, verify and validate the specification, and rapidly prototype the application. The formal specification of the system is based on a Symbol Relation (SR) grammar. Such a multidimensional grammar is particularly attractive since it can describe the spatial and temporal aspects of the application. The specification is converted into TAOML, an extension of HTML.

The structure of the multimedia application development system is shown in figure 1 below. It mainly consists of two tools. The Formal Specification Tool allows a specification of the MSS to be created. The specification may be either visual or text-based. The specification is then validated using an SR grammar for TAOs. If the specification is valid, the tool generates TAOML and an HTML template for the specified system. The Prototyping Tool includes an IC Builder to create the index cells comprising the knowledge structure of the TAOs. A TAOML interpreter generates HTML code from the TAOML and HTML template and from the information produced by the IC builder. The application generated can then be executed from any web browser working with the distributed IC Manager which controls the active knowledge structure built out of active index cells.

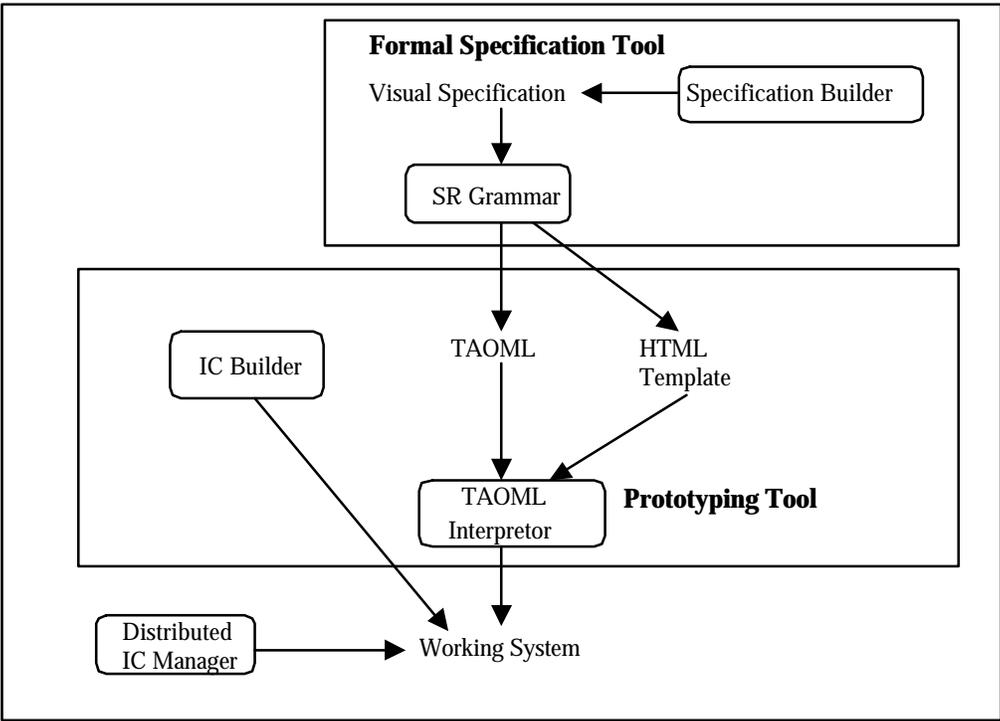


Fig. 1. The structure of the multimedia application development environment

The organization of the rest of this paper is as follows. The following section reviews the TAO concepts as presented in references [5] and [17]. The TAOML extension of HTML is then introduced. This language allows TAO-based systems to be executed from standard web browsers. Section 3 presents a grammatical approach to the formal specification of multimedia applications. Existing multidimensional grammars are analyzed. The Symbol Relation Boundary Grammar (SR Boundary) is chosen as a sufficiently powerful model. The use of this grammar to guide a syntax-directed editor to create the MSS is discussed. A grammar for TAOs is given in section 4. Parsers for this grammar are also discussed. The limitations which must be

imposed on the MSS in order to have an efficient (non-exponential) grammar are given. An attributed form of the grammar is introduced in order to associate static knowledge with the MSS. Discussion and future research are given in section 5. The complete grammar for TAOs is given in appendix A, while the attributed form of the grammar is given in appendix B.

2. Teleaction Objects

The use of grammatical formalisms inside of multimedia systems is the most appropriate way to move from a purely manual approach towards an automatic approach [22]. The advantages to be gained by this approach include the possibility of introducing a graphical front-end for TAO construction, automatic grammatical checking for the correctness of the structure generated, and introduction of a syntax-directed editor. Finally, the integration of both hypergraph and IX production in a single TAO construction module that produces the hypergraph with the IX attached.

Teleaction Objects (TAOs) are multimedia objects with an associated hypergraph representing the structure of the multimedia object and a knowledge structure. The knowledge structure allows the TAO to automatically react to certain events [5].

From a structural point of view, a TAO can be divided in two parts: a *hypergraph G* and *knowledge K*.

The structure of the hypergraph G is a graph $G(N,L)$, where N is a set of nodes, and L is a set of links. There are two types of nodes: base nodes and composite nodes. Each node represents a TAO, and each link represents a relation among TAOs and there are the following link types: the *attachment link*, the *annotation link*, the *reference link*, the *location link*, and the *synchronization link*. Base nodes and composite nodes are called *bundled* when they are grouped, thus defining them as a single entity. The nodes which are interior to bundled nodes may not be included in annotation or reference links unless the link is to the exterior bundled node, and there may not be spatial/temporal relations between interior nodes and nodes external to the bundled node.

The knowledge structure K of a TAO is classified in four levels: the *System Knowledge*, the *Environment Knowledge*, the *Template Knowledge*, and the *Private Knowledge*. The knowledge is structured as an *active index (IX)*, which is a set of *index cells (IC)* from an *index cell base (ICB)*. The index cells define the reactions of the TAO to events filtered by the system. An index cell accepts input messages, performs some action, and sends output messages to a group of ICs. The messages sent will depend on the state of the IC and on the input message [7]. An IC may be seen as a kind of finite-state machine [6].

An initial approach to the definition of a multimedia language for TAOs has been given in [7]. The physical appearance of a TAO is described by a *multidimensional sentence*. The language is generated by a grammar whose alphabet contains generalized icons and operators. Formally, a generalized icon is defined as $x=(x_m,x_i)$ where x_m is the meaning of the icon and x_i is the media object. Two functions, materialization and dematerialization, are associated with every generalized icon. The first function derives the object from its meaning: $MAT(x_m)=x_i$; the second derives the meaning, or interpretation, from the object: $DMA(x_i)=x_m$.

The generalized icons [8] are divided into the following categories:

- Icon: (x_m, x_i) , where x_i is an image
- Earcon: (x_m, x_e) , where x_e is a sound
- Ticon: (x_m, x_t) , where x_t is text (the ticon can also be seen as a subtype of icon).
- Micon: (x_m, x_s) , where x_s is a sequence of image icons (motion icon)

- Vicon: (x_m, x_v) , where x_v is a video clip (video icon)
- Multicon: (x_m, x_c) , where x_c is a multimedia sentence (composite icon).

The generalized icons are represented by nodes in the hypergraph while operators are represented by links.

Example 1: Let us consider a kiosk in a train station which presents tourist information about the surrounding area. The opening screen of the presentation played by the kiosk displays an informative message inviting potential users to touch the screen. When a tourist touches the screen a video begins to play along with some background music. Beneath the video, a sequence of text messages describing the video is displayed. At the end of the video, a screen displaying information on local hotels is visualized. After a short time, the initial message is displayed again and the system waits for the next tourist. In figure 2, we show the hypergraph part of the TAO.

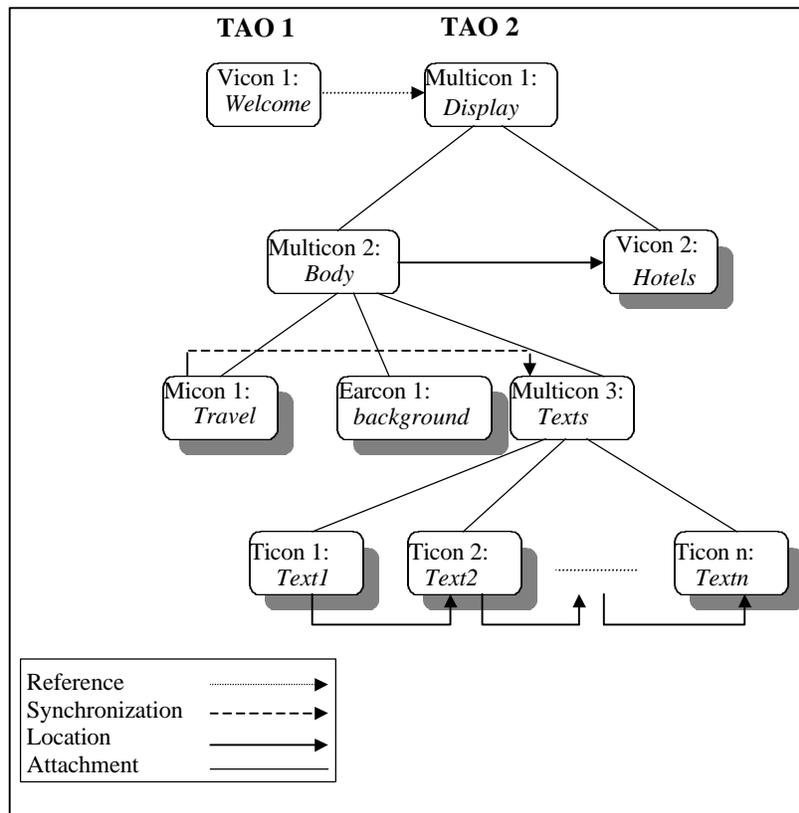


Fig. 2. Kiosk hypergraph

The ICs are attached to the hypergraph to define the actions of the TAO as shown in figure 3. Index cell IC1 is attached to TAO1, *Welcome*, while index cell IC2 is attached to TAO2, *Display*. IC1 is in the state S_0 until the user intervenes with an action by touching the screen. When the message is filtered by the system, it reaches IC1, which passes into state S_1 and sends a message to IC2. IC2 passes from the dead state to the active state. It remains in this state until its lifetime is finished or until the user intervenes causing a stop

action. This action will cause IC2 to return to the dead state and a message to be sent to IC1 which returns to state S0 where it waits for a new "touch" message.

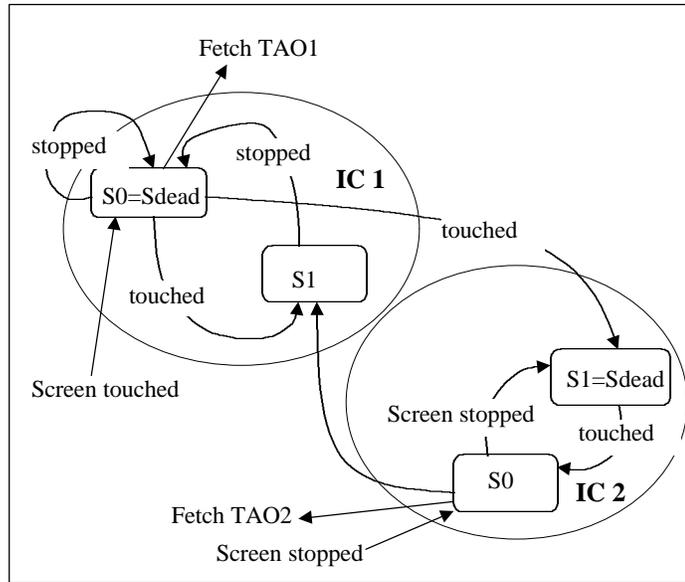


Fig. 3. The ICs of the kiosk TAO

2.1. TAOML

To prototype a distributed multimedia application, each component of the application can be realized as an IC associated with a TAO-enhanced html page. Given a TAO-enhanced html page, we can use an interpreter to read this page, abstract the necessary TAO data structure and generate the normal html page for the browser. Therefore no matter which browser is used, the application program can run if this TAO_HTML interpreter is installed in advance on the web server. This can give some security guarantees. The user can also choose a favorite browser. Furthermore, if in the future HTML is out of fashion, the user need only update the interpreter to output another language. The other parts of the application will not be affected. In this section, we describe the TAO enhanced html named TAOML.

In order to use TAO_HTML, or TAOML, to define a TAO, the structure of a TAO is extended. The new form of the TAO has the following attributes: tao_name, tao_type, p_part, links, ics and sensitivity. These attributes are described below.

- 'tao_name' is the name of the TAO, which is a unique identifier of each TAO.
- 'tao_type' is the media type of TAO - image, text, audio, motion graphs, video or mixed.
- 'p_part' is the physical part of a TAO (see the definition of generalized icon in [8]). To implement this in the context of TAO_HTML, 'p_part' here can be denoted by an HTML template which indicates the appearance of an HTML page. Templates define the fundamental display element and location arrangement. For example, if the TAO is of image type, the template will just contain an HTML statement to introduce an

image. If the TAO is of mixed type, the template will define some common parts and leave some space to insert the elements that are specific to the TAO.

- 'links' are the links to another TAO. A link has attributes 'link_type' and 'link_obj'. 'link_type' is either relational (spatial or temporal) or structural (COMPOSED OF). In the context of TAO_HTML, a spatial link describes visible relationship between subobjects inside one mixed object. For example, a mixed TAO 1 contains an image TAO2 and a text TAO3; then TAO1 has a spatial link with both TAO2 and TAO3. A temporal link usually refers to an invisible object that is not a display element, but whose activation time is influenced by the other TAO. A structural link relates one TAO with another dynamically via user input or external input. For example, the user clicking a button in TAO1 will invoke another page TAO2; in this case there is a structural link from TAO1 to TAO2.

- 'ic' is the associated index cell. The flag is "old" if the ic already exists, or "new" if the ic is to be created. The ic type, ic_id list, message type and message content can either be specified, or input at run-time by the user (indicated by a question mark in the input string). A corresponding HTML input form will be created so that the user can send the specified message to the ics. For further details on the meaning of the attributes of the index cells, see [6].

- 'sensitivity' indicates whether this object is location-sensitive, time-sensitive, content-sensitive or none-sensitive. Then the same object can have different appearances or different functionalities according to the sensitivity. For example, if TAO1 is content-sensitive, it is red when being contained in TAO2 while it is green when being activated by TAO3 via a button. The detailed meaning of sensitivity should be defined by the user according to the requirements of an application. In the newest generation of browsers, sensitivity can be implemented using style sheets.

- 'database' specifies the database that this TAO can access and/or manipulate.

BNF form for TAO_HTML. The formal definition of the TAO_HTML language is given below.

TAO_HTML ::= <TAO> TAO_BODY </TAO>

TAO_BODY ::= NAME_PART TYPE_PART P_PART LINK_PART IC_PART SENSI_PART DATA_PART

NAME_PART ::= <TAO_NAME> "name" </TAO_NAME>

TYPE_PART ::= <TAO_TYPE> TYPE_SET </TAO_TYPE>

TYPE_SET ::= **image | text | audio | motion_graph | video | mixed**

P_PART ::= <TAO_TEMPLATE> "template_name" </TAO_TEMPLATE>

LINK_PART ::= empty | <TAO_LINKS> LINK_BODY </TAO_LINKS> LINK_PART

LINK_BODY ::= name = "link_name", type = LINK_TYPE, obj = "link_obj"

LINK_TYPE ::= spatial | temporal | structural

IC_PART ::= empty | <TAO_IC> flag=FLAG ic_type="a_string" ic_id_list="a_string" cgi_pgm="a_string" message_type="a_string" content="a_string" </TAO_IC>

FLAG ::= old | new

SENSI_PART ::= empty | <TAO_SENSI> SENSITIVITY </TAO_SENSI>

SENSITIVITY ::= location | content | time

DATA_PART ::= empty | <TAO_DATA> "database_name" </TAO_DATA>

In the template of a TAO, in addition to the normal HTML tags and definitions, there is a special TAO tag for a link relation with other TAOs. It is defined as:

<TAO_REL> "link_name" </TAO_REL>

TAO_HTML Interpreter Algorithm. The TAO_HTML Interpreter translates the TAOML pages into HTML pages so that the user interface is easily implemented using a standard web browser. The TAO_HTML Interpreter is now presented in pseudo-code.

```
procedure interpreter(string TAOname)
begin
  open TAO definition file
  while (not end of file) do
    begin
      read one line from the file
      recognize tag
      get tag information
      store in data structure TAO_struct
    end
  call template_parser(TAO_struct)
end
```

```
procedure template_parser(TAO_structure TAO_struct)
begin
  if (IC_PART is specified) then
    output HTML statements to create a form to accept user's input and
    send message to the ic's through IC_Manager
  if (template file exists) then
    open template file
    while (not end of file) do
```

```

begin
  read one line from the file
  if (not <TAO_rel> tag) then
    output html text
  else
    begin
      get link_name from the <TAO_rel> tag
      search in the TAO_structure for link_name
      if (a link structure is found with
        the same link_name) then
        begin
          get link_type and link_TAO_name
          if (link_type=structural) then
            insert <a href..> link in template
            to link with link_TAO_name
          elsif (link_type=spatial) then
            /* insert template of link_TAO_name */
            call interpreter(link_TAO_name)
          end /* if */
        end /* else */
      end /* while */
    end /* procedure */

```

3. Formal Specification - The Grammatical Approach

Formal methods have been proposed as a means for software system designers to assure that a system's requirements accurately reflect the users' requirements and that an implementation is an accurate realization of the design. For these reasons, formal methods provide added reliability to a system. Many researchers claim that formal methods also result in reduced costs since much of the cost of software is a result of imprecision and ambiguity in requirements analysis which necessitates increased testing and maintenance [19]. Formal methods allow a software design to be mathematically modeled and analyzed. A notation for formal specification of a system is provided which can be used to reason about a system in a rigorous manner. In spite of the gains to be realized by adopting formal methods, industrial adoption has been slow. One widely cited impediment to the adaptation of formal methods is the lack of supporting tools.

Due to the complex nature of multimedia applications, they are prime candidates for the application of formal methods. In order to best serve the needs of developers of such applications, we have considered methods specialized for multimedia applications and have settled on a grammatical approach for modeling. Such an approach is well suited to model the hierarchical structure and complex relations of the TAOs composing our applications. It is also possible to implement tools for the construction, manipulation and analysis of grammatical structures, in this way overcoming one of the most serious impediments to the adaptation of formal methods. These observations are the basis for our selection of a grammatical approach to formal specification of multimedia applications. Formal specification stands as one of the foundations of our approach to the design of multimedia applications, along with the TAO framework and prototyping tools.

Much research has been conducted on multimedia systems and on the interaction between multimedia objects and users [3], however few researchers have used a grammatical approach to specify such systems. One who has is Wittenburg [21] who proposed a system based on a relational grammar that allows the automatic presentation of multimedia objects. Certain characteristics distinguish his system from ours; in

particular, Wittenburg's system does not permit interaction between media objects. The user decides the relations between the media objects that are resolved in a phase of constraint solving. The grammar is used to find the correct values for the physical attributes of the objects in a system in which the user may list the relations to derive without giving the values that the attributes must take. It is not clear, however, how much interaction the user may specify. Finally, due to how it is used, the grammar is directly tied to the parser to be used [20]. This limits the generation of multimedia presentations by the system to those that can be analyzed by the parser.

It is important to make the following observation, which also applies to Wittenburg's relational grammars [21, 22, 23], on the relation between visual grammars and parsers. Today many multidimensional grammars having high generative powers have been produced. This is in contrast to the limited recognizing powers of parsers that are penalized by the high computational complexity of multidimensional grammars. While some researchers see this complexity as a limit on the practical utility of multidimensional grammars, we believe that the parser can be avoided by, for example, using syntax-directed editors. General-purpose editor/browsers offer little assistance to the user, while editor/browsers that identify errors and give users the possibility to redo the errors once they have been identified are more useful. Such editors are syntax-directed and can be used to avoid the complexity of the parser [10].

3.1. Multidimensional Grammar

After reviewing the existing grammars, we have excluded the more strict context-free grammar models, like Positional Context-Free and Constraint Multiset Context-Free, because of their limited generative power. In fact, these grammars cannot generate graph languages. Multimedia applications require the handling of complex structures during the parsing phase, therefore a more powerful generative grammar model has to be chosen. However, it is also necessary to avoid increasing the complexity of the parser.

Concerning the complexity of the grammars, a limit on the complexity of parsers of graph grammars has been given by Brandenburg [4] for graph grammars in the *confluence* property. For multidimensional languages, some grammatical derivations that may appear context-free are not since changing the rewrite order of the nonterminals in the derivation can change the final result [12]. Guaranteeing that the result of all grammatical derivations in a language is independent of the rewrite order of the nonterminals guarantees, by definition, the confluence [4]. This property is indispensable for efficient parsers since any order of application of the rules must result in the recognition of the sentences belonging to the language. If the language is not confluent, any parser must evaluate multiple orderings in order to recognize a sentence.

Then we have turned our attention to multidimensional grammars such as Context-Free Positional Grammars [9], Constraint Multiset Grammars [18], and Picture Layout Grammars [13]. These grammars use attributes as an essential part of the parsing algorithm since the values of the attributes are crucial for syntactic analysis.

On the other hand, the role of the attributes in the formal structure of a multimedia presentation is primarily to attach semantic knowledge to the grammar model. The knowledge we need to attach may contain information dependent on the application domain as well as information about semantic actions to be triggered. Such knowledge requires a variety of attributes which should also contribute to semantic analysis. This motivation leads us to SR Grammars [12].

3.2. Symbol Relation Grammars

A common grammatical approach for the description of multidimensional languages uses rewriting mechanisms to generate sentences in the language (e.g. Constraint Multiset Grammars [18], and Picture Layout Grammars [13]). The SR Grammar is one of these grammars. In the SR Grammar formalism [12], a sentence is viewed as a set of symbol occurrence (s-items) and a set of relation items over symbol occurrences (r-items).

The main feature of SR grammars is that the derivation of a sentence is performed by rewriting both symbol occurrences and relation items by means of simple context-free style rules. More precisely, during a derivation step a symbol occurrence X^0 in a sentence S^1 is replaced by a sentence S^2 , according to a rewriting rule of the form $X^0 \rightarrow S^2$, called an s-item production (s-production). After X^0 has been rewritten, the replacement of the set of r-items involving X^0 is performed through r-item rewriting rules (r-productions) of the form $r(X^0, Y^1) \rightarrow R$, where R is a set of r-items relating Y^1 to s-items in S^2 .

In [12] it has been shown how the notion of attribute context-free grammars may be applied to SR Grammars to implement semantic actions in the language and a boundary version of the SR grammar has been proposed. The Boundary SR Grammar has the confluence property and thus a lower computational complexity. An efficient parser has been given [12] for confluent languages, which have the connection and limited degree properties, where this last property means that the number of relations that tie one object to another is limited.

4. A Boundary SR Grammar for the TAO Hypergraph Structure

In this section we describe a Boundary SR Grammar (BSRG) capable of generating the hypergraph structure of the TAO. The grammar is completely general since it does not identify a specific set of relations to be used to construct the TAO. Rather, it permits the instantiation of an arbitrary number of relations since the grammar is defined on base relation types. We identify the following base relation types: spatial; temporal; annotation; reference to the external environment; reference from the external environment. This permits us to use the grammar in various ways, for example, as a module which drives a syntax-directed editor with phases for link creation, assignment of a name to a link, and assignment of a semantic meaning to a link. The grammar is easily specialized for a group of relations for a particular domain. This is possible due to the rewriting of the relations. We exploit this mechanism by having relations represented by non-terminals, which are rewritten with terminal relations only when both terminal nodes involved are reached.

In the subsequent phase of semantic analysis it will be necessary to consider the meaning of the relations, and therefore we introduce an attributed form of the BSRG in which extra information is encapsulated in the attributes attached to the relation.

4.1. A Boundary SR Grammar

The complete version of the BSRG which generates a language that is the set of legal hypergraphs of a TAO is given in appendix A. The most important rules for the construction of a TAO are briefly described in the following. In order to describe, in a synthetic way, the productions of the grammar, we will use the symbols z , x to represent, respectively, the elements of the following sets:

- $z \in \{\text{icon, earcon, vicon, ticon, micon}\}$
- $x \in \{\text{icon, vicon, ticon, micon}\}$

- the initial production either directly produces a terminal node or a composite node and a non-terminal node connected by the attachment relation. The only attachment relations derivable are between a multicon (i.e. a composite node) and its children:

$$1: S^0 \rightarrow \langle \{ \text{multicon}^1, A^1 \} \{ \text{attach}(\text{multicon}^1, A^1) \} \rangle$$

$$17: S^0 \rightarrow \langle \{ x^1 \} \{ \emptyset \} \rangle \quad x \in \{ \text{icon}, \text{vicon}, \text{ticon}, \text{micon} \}$$

- it is possible to derive a reference link to and from the external environment i.e.

$$18: S^0 \rightarrow \langle \{ x^1, \text{EXT}^1 \} \{ \text{reference}(x^1, \text{EXT}^1) \} \rangle$$

where $z \in \{ \text{icon}, \text{earcon}, \text{vicon}, \text{ticon}, \text{micon} \}$ and EXT represents the external environment. Productions 2-16, 18-24, 58-67 describe the external reference to the TAO.

- The annotation relations have as a parent node any base or composite node, but must have as child node the composite node of a new TAO annotating the preceding node, i.e.:

$$43: A^0 \rightarrow \langle \{ x^1, A^1, S^1 \} \{ \text{rel}(x^1, A^1) \text{ annotation}(x^1, S^1) \} \rangle$$

See productions 43-47, 51-57 and 64-67.

- The spatial and temporal relations are derived via the *rel* relation which is rewritten when a terminal is involved. Productions 25-26, 29, 35-36, 39, 43-44, 47, 53-54, 56-57 produce the *rel* relation; i.e.:

$$29: A^0 \rightarrow \langle \{ x^1, A^1 \} \{ \text{rel}(x^1, A^1) \text{ rel}(A^1, x^1) \} \rangle$$

which can be rewritten by using the following rewriting rules:

$$R64: \text{rel}(x^0, A^0) \rightarrow [25,26,27,28,29,43,44,45,46,47] \{ y(x^0, x^1) \}$$

$$R65: \text{rel}(A^0, x^0) \rightarrow [25,26,27,28,29,43,44,45,46,47] \{ y(x^1, x^0) \}$$

where $x \in \{ \text{icon}, \text{vicon}, \text{ticon}, \text{micon}, \text{multicon} \}$ and $y \in \{ \text{synchronization}, \text{location} \}$

- For the earcon, rewritings with spatial relations are forbidden (see productions 30-34, 48-52); i.e.:

$$34: A^0 \rightarrow \langle \{ \text{earcon}^1, A^1 \} \{ \text{synchronization}(\text{earcon}^1, A^1) \text{ synchronization}(A^1, \text{earcon}^1) \} \rangle$$

$$R66: \text{rel}(x^0, A^0) \rightarrow [30,31,32,33,34,48,49,50,51,52] \{ \text{synchronization}(x^0, \text{earcon}^1) \}$$

$$R67: \text{rel}(A^0, x^0) \rightarrow [30,31,32,33,34,48,49,50,51,52] \{ \text{synchronization}(\text{earcon}^1, x^0) \}$$

- the spatial, temporal and reference relations are derived at a high level of derivation. These may be duplicated, rewritten and located in any point of a TAO except when we wish to derive a bundled node. In the case of a bundled node we use productions 40-41:

$$40: A^0 \rightarrow \langle \{ B^1 \} \{ \emptyset \} \rangle$$

$$41: B^0 \rightarrow \langle \{ A^1 \} \{ \emptyset \} \rangle$$

There are no rewriting rules for the relations after the application of the above productions. As a consequence, the sentential forms of the language which we obtain do not have reference links, location links, or synchronization links which involve nodes both external to and internal to the bundle.

The sequence of derivations steps used to generate the TAO of example 1, is shown in figure 4. In the example the nonterminal symbols of each derivation step, which need to be rewritten later, are shown in bold. The relations which involve these symbols and therefore have not yet been rewritten, are also in bold. At each step of the derivation, next to the \Rightarrow symbol, we indicate the s-production and the r-production(s) which have

been involved in the derivation step.

TAO 1 :

S^0

$\Rightarrow 18 \langle \{vicon^1, \mathbf{EXT}^1\} \{reference(vicon^1, \mathbf{EXT}^1)\} \rangle$

$\Rightarrow 58, R156 \langle \{vicon^1, ext^1\} \{reference(vicon^1, ext^1)\} \rangle$

TAO 2:

S^0

$\Rightarrow 3 \langle \{multicon^1, \mathbf{A}^1, \mathbf{EXT}^1\} \{attach(multicon^1, \mathbf{A}^1) reference(\mathbf{EXT}^1, multicon^1)\} \rangle$

$\Rightarrow 58, R156 \langle \{multicon^1, \mathbf{A}^1, ext^1\} \{attach(multicon^1, \mathbf{A}^1)\} \{reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 35, R1 \langle \{multicon^1, \{multicon^2, \mathbf{A}^2, \mathbf{A}^3\}, ext^1\} \{rel(multicon^2, \mathbf{A}^2) attach(multicon^2, \mathbf{A}^3) attach(multicon^1, \mathbf{A}^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 28, R64, R25 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \mathbf{A}^3\}, ext^1\} \{rel(multicon^2, vicon^2) attach(multicon^2, \mathbf{A}^3) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 40, R23 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{\mathbf{B}^1\}\}, ext^1\} \{rel(multicon^2, vicon^2) attach(multicon^2, \mathbf{B}^1) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 41, R24 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{\mathbf{A}^4\}\}, ext^1\} \{rel(multicon^2, vicon^2) attach(multicon^2, \mathbf{A}^4) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 36, R1 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{multicon^3, \mathbf{A}^5, \mathbf{A}^6\}\}, ext^1\} \{rel(\mathbf{A}^5, multicon^3) attach(multicon^3, \mathbf{A}^6) rel(multicon^2, vicon^2) attach(multicon^2, \mathbf{A}^5) attach(multicon^2, multicon^3) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 42, R61, R19 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{multicon^3, \{\mathbf{A}^7\}, \mathbf{A}^6\}\}, ext^1\} \{rel(\mathbf{A}^7, multicon^3) rel(\mathbf{A}^7, multicon^3) attach(multicon^3, \mathbf{A}^6) rel(multicon^2, vicon^2) attach(multicon^2, \mathbf{A}^7) attach(multicon^2, multicon^3) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 25, R65, R63, R28 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{multicon^3, \{micon^1, \mathbf{A}^8\}, \mathbf{A}^6\}\}, ext^1\} \{rel(micon^1, \mathbf{A}^8) location(micon^1, multicon^3) rel(\mathbf{A}^8, multicon^3) attach(multicon^3, \mathbf{A}^6) rel(multicon^2, vicon^2) attach(multicon^2, micon^1) attach(multicon^2, \mathbf{A}^8) attach(multicon^2, multicon^3) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 33, R66, R67, R25 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{multicon^3, \{micon^1, \{earcon^1\}\}, \mathbf{A}^6\}\}, ext^1\} \{synchronization(micon^1, earcon^1) location(micon^1, multicon^3) synchronization(earcon^1, multicon^3) attach(multicon^3, \mathbf{A}^6) rel(multicon^2, vicon^2) attach(multicon^2, micon^1) attach(multicon^2, earcon^1) attach(multicon^2, multicon^3) attach(multicon^1, vicon^2) attach(multicon^1, multicon^2) reference(ext^1, multicon^1)\} \rangle$

$\Rightarrow 25, R28 \langle \{multicon^1, \{multicon^2, \{vicon^2\}, \{multicon^3, \{micon^1, \{earcon^1\}\}, \{ticon^1, \mathbf{A}^9\}\}, ext^1\} \{rel(ticon^1, \mathbf{A}^9)\} \rangle$

synchronization(micon¹, earcon¹) location(micon¹, multicon³) synchronization(earcon¹, multicon³) attach(multicon³, ticon¹)
attach(multicon³, **A**⁹) rel(multicon², vicon²) attach(multicon², micon¹) attach(multicon², earcon¹) attach(multicon², multicon³)
attach(multicon¹, vicon²) attach(multicon¹, multicon²) reference(ext¹, multicon¹)>>

⇒25,R64,R28 <{multicon¹, {multicon², {vicon²}}, {multicon³, {micon¹, {earcon¹}}, {ticon¹, {ticon², **A**¹⁰}}}, ext¹} {**rel**(ticon²,
A¹⁰) synchronization(ticon¹, ticon²) synchronization(micon¹, earcon¹) location(micon¹, multicon³) synchronization(earcon¹,
multicon³) attach(multicon³, ticon¹) attach(multicon³, ticon²) **attach**(multicon³, **A**¹⁰) rel(multicon², vicon²) attach(multicon²,
micon¹) attach(multicon², earcon¹) attach(multicon², multicon³) attach(multicon¹, vicon²) attach(multicon¹, multicon²)
reference(ext¹, multicon¹)}>>

.....

⇒25,R64,R28 <{multicon¹, {multicon², {vicon²}}, {multicon³, {micon¹, {earcon¹}}, {ticon¹, {ticon², {.....{ticonⁿ⁻¹, **A**ⁿ⁻¹⁺⁸}.....},
ext¹} {**rel**(ticonⁿ⁻¹, **A**ⁿ⁻¹⁺⁸) synchronization(ticonⁿ⁻², ticonⁿ⁻¹) synchronization(ticon², ticon³) synchronization(ticon¹, ticon²)
synchronization(micon¹, earcon¹) location(micon¹, multicon³) synchronization(earcon¹, multicon³) attach(multicon³, ticon¹)
attach(multicon³, ticon²) attach(multicon³, ticonⁿ⁻²) **attach**(multicon³, **A**ⁿ⁻¹⁺⁸) rel(multicon², vicon²) attach(multicon²,
micon¹) attach(multicon², earcon¹) attach(multicon², multicon³) attach(multicon¹, vicon²) attach(multicon¹, multicon²)
reference(ext¹, multicon¹)}>>

⇒28,R64,R26 <{multicon¹, {multicon², {vicon²}}, {multicon³, {micon¹, {earcon¹}}, {ticon¹, {ticon², {.....{ticonⁿ⁻¹, ticonⁿ}.....},
ext¹} {rel(ticonⁿ⁻¹, ticonⁿ) synchronization(ticonⁿ⁻², ticonⁿ⁻¹) synchronization(ticon², ticon³) synchronization(ticon¹, ticon²)
synchronization(micon¹, earcon¹) location(micon¹, multicon³) synchronization(earcon¹, multicon³) attach(multicon³, ticon¹)
attach(multicon³, ticon²) attach(multicon³, ticonⁿ⁻²) attach(multicon³, ticonⁿ⁻¹) attach(multicon³, ticonⁿ) rel(multicon²,
vicon²) attach(multicon², micon¹) attach(multicon², earcon¹) attach(multicon², multicon³) attach(multicon¹, vicon²)
attach(multicon¹, multicon²) reference(ext¹, multicon¹)}>>

Fig. 4. Derivation steps of kiosk TAO

1.2. Parsing

As stated in section 3.1, the confluence property is important for multidimensional languages since if the language satisfies this property an efficient parser for the language can be produced. A Boundary SR grammar must satisfy two constraints in order to be confluent - the graphs generated by the language must be connected and each node must have a limited degree (i.e. the number of links from each node must be less than or equal to some constant).

The hypergraphs generated by the Boundary SR grammar for TAOs given in Appendix A are connected since the hypergraph is uniquely given by the derivation tree with root S, the start symbol. If we have two TAOs, these TAOs may be connected by a reference link. This reference link is the point of connectivity between the two TAOs.

The limited connectivity property is also satisfied, even if we are not able to give a priori a limit. It is reasonable to assume (since it doesn't limit the type of TAOs we wish to generate) that a node is linked only to its neighbors. Further, there is a constant number of link types. Therefore, even if a node is connected to

its neighbors via all link types there is not a linear degree of connectivity. This limitation drastically reduces the size of the language, but it does not disallow the sentences (i.e. TAOs) that we are interested in.

5. Semantic Extension of TAO using Attribute SR Grammars

Teleaction objects consist of a hypergraph representing the interface of a multimedia application with an associated knowledge structure. We have shown how the hypergraph structure can be generated by an SR grammar. The knowledge structure is an active index and is created by using the IC builder tool. It is possible to extend the SR grammar for TAOs to include the associated knowledge as semantic actions associated with the grammar. An extension of the SR grammar which does this was proposed in [11]. The extended SR grammar is an Attribute SR Grammar which associates a set of inherited and/or synthesized attributes with the non-terminal symbols of V_N and with the symbols of the relations of V_R , associating evaluation rules with the s- and r-productions. This permits the use of different knowledge in different contexts, using the context-sensitive generative power of the derivations to pass the attributes.

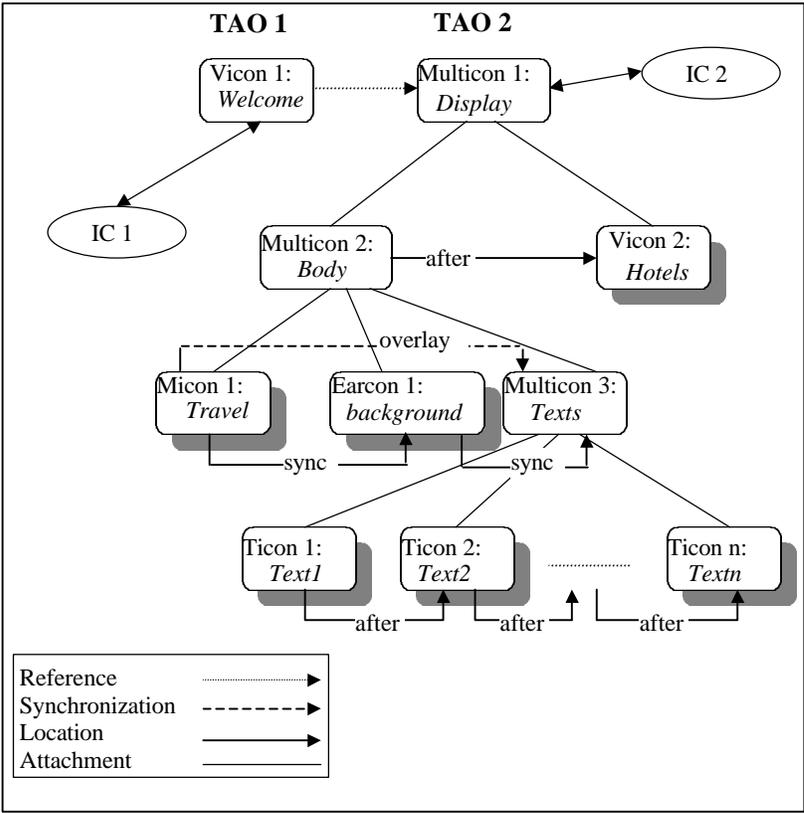


Fig. 5. Hypergraph with attached knowledge

5.1. Environmental adaptability

Since the grammatical model provides a variety of relations, the knowledge will provide the correct routines for materialization/interpretation of these relations. We can give the same names to different routines that work with diverse media types and, by attribute passing, let the correct routines reach the terminals. This approach allows compatibility with diverse multimedia environments in which a relation may have many meanings.

Knowledge is expressed as references to an area of memory of the distributed ICs. The correct knowledge base will be loaded in this area of memory. Since we separate the construction phase of the environment from the construction phase of a TAO, multiple TAO systems may be constructed in the same environment. Furthermore, diverse environments may be supported by loading an environment-specific knowledge base. For this, it is sufficient to use only inherited attributes [1]. Given the limited nature of their use, a dependence among the attributes of parents and children only is assured, thus assuring the acyclic nature of the dependence graph and the efficiency of the calculations (a top down visit using the hypergraph attachments is sufficient, or from the generative point of view, a top down visit of the derivation tree).

The attribute scheme for a TAO is given in [2].

6. Conclusions

We have presented the basis for a principled approach to the production of distributed multimedia applications. The unifying principle for our approach is the Teleaction Object. TAOML, an extension of HTML has been introduced to allow distributed multimedia applications to be prototyped using standard web browsers as a front-end and the distributed IC manager to manage the knowledge associated with the application. A boundary SR grammar has been introduced to allow for the formal specification of TAOs. The interpreter for TAOML as well as the distributed IC manager and a graphical front-end for specifying TAOs have been developed. In the future, we will develop a syntax-directed editor based on the grammar. This editor will produce TAOML output via semantic actions. This output will then be fed to the interpreter, providing a unified approach to application development. We also plan to investigate the use of the formal specification to prove properties of the application.

Appendix A: The Boundary Symbol Relation Grammar for the TAO

The BSRG G for the TAO is defined as follow. $G = (V_N, V_T, V_R, S, P, R)$ where S is the start symbol, the set of nonterminals is $V_N = \{S, A, B, EXT\}$, the set of terminals is $V_T = \{icon, vicon, earcon, ticon, micon, multicon, ext\}$ and the set of relations is $V_R = \{rel, attach, annotation, synchronization, location, reference\}$. The terminal *ext* represent the icons that have an external reference to or from them.

Notation: *in order to avoid duplication of productions which differ by just one terminal symbol, we introduce the symbols x, z, t, y to represent, respectively, the symbols of the following sets:*

$x\hat{I} \{icon, vicon, ticon, micon\}$

$z\hat{I} \{icon, earcon, vicon, ticon, micon\}$

$t\hat{I} \{icon, earcon, vicon, ticon, micon, multicon\}$

$y\hat{I} \{synchronization, location\}$

- 31: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1 \} \{ \text{synchronization} (A^1, \text{earcon}^1) \} \rangle$
32: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1 \} \{ \emptyset \} \rangle$
33: $A^0 \rightarrow \langle \{ \text{earcon}^1 \} \{ \emptyset \} \rangle$
34: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1 \} \{ \text{synchronization} (\text{earcon}^1, A^1) \text{synchronization} (A^1, \text{earcon}^1) \} \rangle$
- 35: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2 \} \{ \text{rel}(\text{multicon}^1, A^1) \text{attach}(\text{multicon}^1, A^2) \} \rangle$
36: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2 \} \{ \text{rel}(A^1, \text{multicon}^1) \text{attach}(\text{multicon}^1, A^2) \} \rangle$
37: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2 \} \{ \text{attach}(\text{multicon}^1, A^2) \} \rangle$
38: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^2 \} \{ \text{attach}(\text{multicon}^1, A^2) \} \rangle$
39: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2 \} \{ \text{rel}(\text{multicon}^1, A^1) \text{rel}(A^1, \text{multicon}^1) \text{attach}(\text{multicon}^1, A^2) \} \rangle$
- 40: $A^0 \rightarrow \langle \{ B^1 \} \{ \emptyset \} \rangle$
41: $B^0 \rightarrow \langle \{ A^1 \} \{ \emptyset \} \rangle$ /*The productions 40 and 41 generate the bundled nodes. In fact, there are not existing r-productions for the relation *rel*, which include these productions. As a consequence, the sentential forms of the language which we obtain do not have reference links, location links, or synchronization links which involve nodes both external to and internal to the bundle. We have r-productions only for the attachment links.*/
42: $A^0 \rightarrow \langle \{ A^1 \} \{ \emptyset \} \rangle$
- 43: $A^0 \rightarrow \langle \{ x^1, A^1, S^1 \} \{ \text{rel} (x^1, A^1) \text{annotation}(x^1, S^1) \} \rangle$
44: $A^0 \rightarrow \langle \{ x^1, A^1, S^1 \} \{ \text{rel} (A^1, x^1) \text{annotation}(x^1, S^1) \} \rangle$
45: $A^0 \rightarrow \langle \{ x^1, A^1, S^1 \} \{ \text{annotation}(x^1, S^1) \} \rangle$
46: $A^0 \rightarrow \langle \{ x^1, S^1 \} \{ \text{annotation}(x^1, S^1) \} \rangle$
47: $A^0 \rightarrow \langle \{ x^1, A^1, S^1 \} \{ \text{rel} (x^1, A^1) \text{rel} (A^1, x^1) \text{annotation}(x^1, S^1) \} \rangle$
- 48: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1, S^1 \} \{ \text{synchronization} (\text{earcon}^1, A^1) \text{annotation}(\text{earcon}^1, S^1) \} \rangle$
49: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1, S^1 \} \{ \text{synchronization} (A^1, \text{earcon}^1) \text{annotation}(\text{earcon}^1, S^1) \} \rangle$
50: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1, S^1 \} \{ \text{annotation}(\text{earcon}^1, S^1) \} \rangle$
51: $A^0 \rightarrow \langle \{ \text{earcon}^1, S^1 \} \{ \text{annotation}(\text{earcon}^1, S^1) \} \rangle$
52: $A^0 \rightarrow \langle \{ \text{earcon}^1, A^1, S^1 \} \{ \text{synchronization} (\text{earcon}^1, A^1) \text{synchronization} (A^1, \text{earcon}^1) \text{annotation}(\text{earcon}^1, S^1) \} \rangle$
- 53: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2, S^1 \} \{ \text{rel}(\text{multicon}^1, A^1) \text{attach}(\text{multicon}^1, A^2) \text{annotation}(\text{multicon}^1, S^1) \} \rangle$
54: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2, S^1 \} \{ \text{rel}(A^1, \text{multicon}^1) \text{attach}(\text{multicon}^1, A^2) \text{annotation}(\text{multicon}^1, S^1) \} \rangle$
55: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2, S^1 \} \{ \text{attach}(\text{multicon}^1, A^2) \text{annotation}(\text{multicon}^1, S^1) \} \rangle$
56: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^2, S^1 \} \{ \text{rel}(\text{multicon}^1, A^2) \text{annotation}(\text{multicon}^1, S^1) \} \rangle$
57: $A^0 \rightarrow \langle \{ \text{multicon}^1, A^1, A^2, S^1 \} \{ \text{rel}(\text{multicon}^1, A^1) \text{rel}(A^1, \text{multicon}^1) \text{attach}(\text{multicon}^1, A^2) \text{annotation}(\text{multicon}^1, S^1) \} \rangle$
- 58: $\text{EXT}^0 \rightarrow \langle \{ \text{ext}^1 \} \{ \emptyset \} \rangle$
59: $\text{EXT}^0 \rightarrow \langle \{ \text{EXT}^1 \} \{ \emptyset \} \rangle$
- 60: $S^0 \rightarrow \langle \{ \text{earcon}^1 \} \{ \emptyset \} \rangle$
61: $S^0 \rightarrow \langle \{ \text{earcon}^1, \text{EXT}^1 \} \{ \text{reference}(\text{earcon}^1, \text{EXT}^1) \} \rangle$
62: $S^0 \rightarrow \langle \{ \text{earcon}^1, \text{EXT}^1 \} \{ \text{reference}(\text{EXT}^1, \text{earcon}^1) \} \rangle$
63: $S^0 \rightarrow \langle \{ \text{earcon}^1, \text{EXT}^1 \} \{ \text{reference}(\text{earcon}^1, \text{EXT}^1) \text{reference}(\text{EXT}^1, \text{earcon}^1) \} \rangle$

- R64: $\text{rel}(x^0, A^0) \rightarrow [25,26,27,28,29,43,44,45,46,47] \{y(x^0, x^1)\}$
R65: $\text{rel}(A^0, x^0) \rightarrow [25,26,27,28,29,43,44,45,46,47] \{y(x^1, x^0)\}$
- R66: $\text{rel}(x^0, A^0) \rightarrow [30,31,32,33,34,48,49,50,51,52] \{\text{synchronization}(x^0, \text{earcon}^1)\}$
R67: $\text{rel}(A^0, x^0) \rightarrow [30,31,32,33,34,48,49,50,51,52] \{\text{synchronization}(\text{earcon}^1, x^0)\}$
- R68: $\text{rel}(x^0, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{\text{rel}(x^0, A^2)\}$
R69: $\text{rel}(A^0, x^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{\text{rel}(A^2, x^0)\}$
- R70: $\text{rel}(x^0, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{y(x^0, \text{multicon}^1)\}$
R71: $\text{rel}(A^0, x^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{y(\text{multicon}^1, x^0)\}$
- R72: $\text{rel}(x^0, A^0) \rightarrow [43,44,45,46,47,53,54,55,56,57] \{\text{rel}(x^0, S^1)\}$
R73: $\text{rel}(A^0, x^0) \rightarrow [43,44,45,46,47,53,54,55,56,57] \{\text{rel}(S^1, x^0)\}$
- R74: $\text{synchronization}(\text{earcon}^0, A^0) \rightarrow [25,26,27,28,29,30,31,32,33,34,43,44,45,46,47,48,49,50,51,52] \{\text{synchronization}(\text{earcon}^0, z^1)\}$
R75: $\text{synchronization}(A^0, \text{earcon}^0) \rightarrow [25,26,27,28,29,30,31,32,33,34,43,44,45,46,47,48,49,50,51,52] \{\text{synchronization}(z^1, \text{earcon}^0)\}$
R76: $\text{synchronization}(\text{earcon}^0, A^0) \rightarrow [25,26,27,29,30,31,32,34,35,36,37,39,42,43,44,45,47,48,49,50,52,53,54,55,56,57] \{\text{synchronization}(\text{earcon}^0, A^1)\}$
R77: $\text{synchronization}(A^0, \text{earcon}^0) \rightarrow [25,26,27,29,30,31,32,34,35,36,37,39,42,43,44,45,47,48,49,50,52,53,54,55,56,57] \{\text{synchronization}(A^1, \text{earcon}^0)\}$
- R78: $\text{synchronization}(\text{earcon}^0, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{\text{synchronization}(\text{earcon}^0, A^2)\}$
R79: $\text{synchronization}(A^0, \text{earcon}^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{\text{synchronization}(A^2, \text{earcon}^0)\}$
- R80: $\text{synchronization}(\text{earcon}^0, A^0) \rightarrow [48,49,50,51,52,53,54,55,56,57] \{\text{synchronization}(\text{earcon}^0, S^1)\}$
R81: $\text{synchronization}(A^0, \text{earcon}^0) \rightarrow [48,49,50,51,52,53,54,55,56,57] \{\text{synchronization}(S^1, \text{earcon}^0)\}$
- R82: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1)\}$
R83: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^0, \text{multicon}^1)\}$
R84: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^1, \text{multicon}^0)\}$
R85: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^1, \text{multicon}^0)\}$
- R86: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) \text{rel}(\text{multicon}^0, A^1)\}$
R87: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^0, \text{multicon}^1) \text{rel}(\text{multicon}^0, A^1)\}$
R88: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1) y(\text{multicon}^1, \text{multicon}^0) \text{rel}(\text{multicon}^0, A^1)\}$
R89: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{\text{annotation}(\text{multicon}^0, \text{multicon}^1)\}$

- R111: $\text{annotation}(z^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{ \text{annotation}(z^0, \text{multicon}^1) y(z^0, \text{multicon}^1) \text{rel}(A^1, z^0) \text{rel}(z^0, A^1) \}$
- R112: $\text{annotation}(z^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{ \text{annotation}(z^0, \text{multicon}^1) y(\text{multicon}^1, z^0) \text{rel}(A^1, z^0) \text{rel}(z^0, A^1) \}$
- R113: $\text{annotation}(z^0, S^1) \rightarrow [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16] \{ \text{annotation}(z^0, \text{multicon}^1) y(z^0, \text{multicon}^1) y(\text{multicon}^1, z^0) \text{rel}(A^1, z^0) \text{rel}(z^0, A^1) \}$
- R114: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(\text{multicon}^0, x^1) \}$
- R115: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(\text{multicon}^0, x^1) y(\text{multicon}^0, x^1) \}$
- R116: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(\text{multicon}^0, x^1) y(x^1, \text{multicon}^0) \}$
- R117: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(\text{multicon}^0, x^1) y(x^1, \text{multicon}^0) y(\text{multicon}^0, x^1) \}$
- R118: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(\text{multicon}^0, x^1) \}$
- R119: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(\text{multicon}^0, x^1) \text{synchronization}(\text{multicon}^0, \text{earcon}^1) \}$
- R120: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(\text{multicon}^0, x^1) \text{synchronization}(\text{earcon}^1, \text{multicon}^0) \}$
- R121: $\text{annotation}(\text{multicon}^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(\text{multicon}^0, x^1) \text{synchronization}(\text{earcon}^1, \text{multicon}^0) \text{synchronization}(\text{multicon}^0, \text{earcon}^1) \}$
- R122: $\text{annotation}(z^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(z^0, x^1) \}$
- R123: $\text{annotation}(z^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(z^0, x^1) y(x^1, z^0) \}$
- R124: $\text{annotation}(z^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(z^0, x^1) y(z^0, x^1) \}$
- R125: $\text{annotation}(z^0, S^1) \rightarrow [17,18,19,20,21,22,23,24] \{ \text{annotation}(z^0, x^1) y(x^1, z^0) y(z^0, x^1) \}$
- R126: $\text{annotation}(z^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(z^0, \text{earcon}^1) \}$
- R127: $\text{annotation}(z^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(z^0, \text{earcon}^1) \text{synchronization}(\text{earcon}^1, z^0) \}$
- R128: $\text{annotation}(z^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(z^0, \text{earcon}^1) \text{synchronization}(z^0, \text{earcon}^1) \}$
- R129: $\text{annotation}(z^0, S^1) \rightarrow [60,61,62,63,64,65,66,67] \{ \text{annotation}(z^0, \text{earcon}^1) \text{synchronization}(\text{earcon}^1, z^0) \text{synchronization}(z^0, \text{earcon}^1) \}$
- R130: $\text{reference}(\text{ext}, A^0) \rightarrow [40] \{ \text{reference}(B^1, \text{ext}) \}$
- R131: $\text{reference}(B^0, \text{ext}) \rightarrow [41] \{ \text{reference}(A^1, \text{ext}) \}$
- R132: $\text{reference}(\text{ext}, A^0) \rightarrow [25,26,27,29,30,31,32,34,35,36,37,39,42,43,44,45,47,48,49,50,52,53,54,55,57] \{ \text{reference}(\text{ext}, A^1) \}$
- R133: $\text{reference}(\text{ext}, A^0) \rightarrow [42] \{ \text{reference}(\text{ext}, A^1) \text{reference}(\text{ext}, A^1) \}$
- R134: $\text{reference}(\text{ext}, A^0) \rightarrow [25,26,27,28,29,30,31,32,33,34,43,44,45,46,47,48,49,50,51,52] \{ \text{reference}(z^1, \text{ext}) \}$
- R135: $\text{reference}(\text{ext}, A^0) \rightarrow [25,26,27,28,29,30,31,32,33,34,43,44,45,46,47,48,49,50,51,52] \{ \text{reference}(\text{ext}, z^1) \}$
- R136: $\text{reference}(\text{ext}, A^0) \rightarrow [25,26,27,29,30,31,32,34,43,44,45,47,48,49,50,52] \{ \text{reference}(\text{ext}, A^1) \}$

- R137: $\text{reference}(\text{ext}, A^0) \rightarrow [25,26,27,29,30,31,32,34,43,44,45,47,48,49,50,52] \{ \text{reference}(\text{ext}, A^1) \text{reference}(z^1, \text{ext}) \}$
 $\text{reference}(\text{ext}, z^1) \}$
- R138: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{multicon}^1, \text{ext}) \}$
- R139: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{ext}, \text{multicon}^1) \}$
- R140: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,39,53,54,55,57] \{ \text{reference}(\text{ext}, A^1) \text{reference}(\text{multicon}^1, \text{ext}) \}$
- R141: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,39,53,54,55,57] \{ \text{reference}(\text{ext}, A^1) \text{reference}(\text{ext}, \text{multicon}^1) \}$
- R142: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{ext}, A^2) \text{reference}(\text{multicon}^1, \text{ext}) \}$
- R143: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{ext}, A^2) \text{reference}(\text{ext}, \text{multicon}^1) \}$
- R144: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{ext}, A^2) \}$
- R145: $\text{reference}(\text{ext}, A^0) \rightarrow [35,36,37,39,53,54,55,56,57] \{ \text{reference}(\text{ext}, A^1) \text{reference}(\text{ext}, A^2) \}$
- R146: $\text{reference}(A^0, \text{ext}) \rightarrow [25,26,27,28,29,30,31,32,33,34] \{ \text{reference}(z^1, \text{ext}) \}$
- R147: $\text{reference}(A^0, \text{ext}) \rightarrow [42] \{ \text{reference}(A^1, \text{ext}) \text{reference}(A^1, \text{ext}) \}$
- R148: $\text{reference}(A^0, \text{ext}) \rightarrow [25,26,27,29,30,31,32,34,35,36,37,39,42,43,44,45,47,48,49,50,52,53,54,55,57] \{ \text{reference}(A^1, \text{ext}) \}$
- R149: $\text{reference}(A^0, \text{ext}) \rightarrow [25,26,27,29,30,31,32,34,43,44,45,47,48,49,50,52] \{ \text{reference}(A^1, \text{ext}) \text{reference}(z^1, \text{ext}) \}$
- R150: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{multicon}^1, \text{ext}) \}$
- R151: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(A^2, \text{ext}) \}$
- R152: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,38,39,53,54,55,56,57] \{ \text{reference}(\text{multicon}^1, \text{ext}), \text{reference}(A^2, \text{ext}) \}$
- R153: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,39,53,54,55,57] \{ \text{reference}(\text{multicon}^1, \text{ext}) \text{reference}(A^1, \text{ext}) \}$
- R154: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,39,53,54,55,57] \{ \text{reference}(\text{multicon}^1, \text{ext}) \text{reference}(A^1, \text{ext}) \text{reference}(A^2, \text{ext}) \}$
- R155: $\text{reference}(A^0, \text{ext}) \rightarrow [35,36,37,39,53,54,55,57] \{ \text{reference}(A^1, \text{ext}) \text{reference}(A^2, \text{ext}) \}$
- R156: $\text{reference}(t^0, \text{EXT}^0) \rightarrow [58] \{ \text{reference}(t^0, \text{ext}^1) \}$ $t \in \{ \text{multicon}, z \}$
- R157: $\text{reference}(t^0, \text{EXT}^0) \rightarrow [59] \{ \text{reference}(t^0, \text{EXT}^1) \}$
- R158: $\text{reference}(t^0, \text{EXT}^0) \rightarrow [59] \{ \text{reference}(t^0, \text{EXT}^1) \text{reference}(t^0, \text{EXT}^1) \}$
- R159: $\text{reference}(\text{EXT}^0, t^0) \rightarrow [58] \{ \text{reference}(\text{ext}^1, t^0) \}$
- R160: $\text{reference}(\text{EXT}^0, t^0) \rightarrow [59] \{ \text{reference}(\text{EXT}^1, t^0) \}$
- R161: $\text{reference}(\text{EXT}^0, t^0) \rightarrow [59] \{ \text{reference}(\text{EXT}^1, t^0) \text{reference}(\text{EXT}^1, t^0) \}$

Appendix B: The Attributed Boundary Symbol Relation Grammar for the TAO

Given the grammar of Appendix A we will add an inherited attribute $x.K$ to each x terminal in V_T and nonterminal in V_N to represent the knowledge associated with each node of the hypergraph; an attribute

x.name to each nonterminal in V_N and synchronization and location relation in V_R . The inherited attribute represents a reference to the file that will contain the nonterminal or the relation descriptions respectively. We also add the following sets of semantic rules to each of the productions in P.

For production 1 to 8:

$\text{multicon}^1.K := \text{load}(\text{system knowledge})$
 $A^1.K := \text{load}(\text{system knowledge}) \cup S^0.K$ (this semantic rule is not valid for production 1)
 $\text{multicon}^1.name := \text{load}(\text{name})$

For production 17 to 20:

$z^1.K := \text{load}(\text{system knowledge}) \cup \text{load}(\text{private knowledge})$
 $z^1.name^0 := \text{load}(\text{name})$

For production 25 to 34:

$z^1.K := \text{load}(\text{private knowledge}) \cup A^0.K$
 $A^1.K := A^0.K$ (this semantic rule is not valid for production 28 and 33)

For production 35 to 39:

$\text{multicon}^1.K := \text{load}(\text{environment knowledge}) \cup A^0.K$
 $A^1.K := A^0.K$ (this semantic rule is not valid for production 39)
 $A^2.K := \text{load}(\text{environment knowledge}) \cup A^0.K$
 $\text{multicon}^1.name^0 := \text{load}(\text{name})$

For production 40:

$B^1.K := A^0.K$

For production 41:

$A^1.K := B^0.K$

For production 42:

$A^1.K := A^0.K$

For production 9 to 16:

$\text{multicon}^1.K := \text{load}(\text{system knowledge})$
 $A^1.K := \text{load}(\text{system knowledge})$
 $\text{multicon}^1.name^0 := \text{load}(\text{name})$

For production 21 to 24:

$z^1.K := \text{load}(\text{system knowledge}) \cup S^0.K \cup \text{load}(\text{private knowledge})$
 $z^1.name^0 := \text{load}(\text{name})$

For production 44 to 52:

$z^1.K := \text{load}(\text{private knowledge}) \cup A^0.K$
 $A^1.K := A^0.K$
 $z^1.K := \text{load}(\text{name})$

For production 53 to 57:

$$\begin{aligned} \text{multicon}^1.K &:= \text{load}(\text{environment knowledge}) \cup A^0.K \\ A^1.K &:= A^0.K \\ A^2.K &:= \text{load}(\text{environment knowledge}) \cup A^0.K \\ \text{multicon}^1.\text{name}^0 &:= \text{load}(\text{name}) \end{aligned}$$

We will add the attribute **synchronization(aⁱ,a^j).name** and **location(aⁱ,a^j).name** to the *synchronization* and *location* relations to specify the instance that hold between aⁱ and a^j.

We will add the attribute **reference(ext,a^j).name** and **reference(a^j,ext).name** to the *reference* relation to specify the name of the external icon that will be involved in the relation together with a^j.

We will add the attribute **annotation(ext,a^j).name** to the *annotation* relation to specify the name of the icon that will represent the root of the TAO annotated to a^j.

We also add the following sets of semantic rules to each production in **R** that contains the respective r-item:

$$\begin{aligned} y(\text{multicon}^0, \text{multicon}^1).\text{name} &:= \text{load}(\text{name}) \\ y(\text{multicon}^1, \text{multicon}^0).\text{name} &:= \text{load}(\text{name}) \\ y(\text{multicon}^0, x^1).\text{name} &:= \text{load}(\text{name}) \\ y(x^1, \text{multicon}^0).\text{name} &:= \text{load}(\text{name}) \end{aligned}$$
$$\begin{aligned} \text{synchronization}(\text{multicon}^0, \text{earcon}^1).\text{name} &:= \text{load}(\text{name}) \\ \text{synchronization}(\text{earcon}^1, \text{multicon}^0).\text{name} &:= \text{load}(\text{name}) \end{aligned}$$
$$\begin{aligned} y(x^0, x^1).\text{name} &:= \text{load}(\text{name}) \\ y(x^1, x^0).\text{name} &:= \text{load}(\text{name}) \end{aligned}$$
$$\begin{aligned} \text{synchronization}(x^0, \text{earcon}^1).\text{name} &:= \text{load}(\text{name}) \\ \text{synchronization}(\text{earcon}^1, x^0).\text{name} &:= \text{load}(\text{name}) \\ \text{synchronization}(\text{earcon}^0, z^1).\text{name} &:= \text{load}(\text{name}) \\ \text{synchronization}(z^1, \text{earcon}^0).\text{name} &:= \text{load}(\text{name}) \end{aligned}$$
$$\begin{aligned} \text{reference}(z^1, \text{ext}).\text{name} &:= \text{load}(\text{name}) \\ \text{reference}(\text{ext}, z^1).\text{name} &:= \text{load}(\text{name}) \\ \text{reference}(\text{multicon}^1, \text{ext}).\text{name} &:= \text{load}(\text{name}) \\ \text{reference}(\text{ext}, \text{multicon}^1).\text{name} &:= \text{load}(\text{name}) \end{aligned}$$
$$\begin{aligned} \text{annotation}(\text{multicon}^0, \text{multicon}^1).\text{name} &:= \text{multicon}^1.\text{name} \\ \text{annotation}(\text{multicon}^0, z^1).\text{name} &:= z^1.\text{name} \\ \text{annotation}(z^0, \text{multicon}^1).\text{name} &:= \text{multicon}^1.\text{name} \\ \text{annotation}(z^0, z^1).\text{name} &:= z^1.\text{name} \end{aligned}$$

References

1. Aho AV, Sethi R, Ullman JD (1986) Compilers - principles, techniques and tools. Addison-Wesley Publishing Co..
2. Arndt T, Cafiero A, Guercio A (1997) Symbol Relation Grammars for Teleaction Objects. Technical Report, Dipartimento di Informatica ed Applicazioni, University of Salerno
3. Bottoni P, Costabile MF, Levialdi S, Mussio P (1996) Specification of visual languages as means for interaction. AVI '96 Int. Workshop on Theory of Visual Languages, URL: <http://www.cs.monash.edu.au/~berndm/TVL96/tvl96-home.html>
4. Brandenburg FJ (1988) On polynomial time graph grammars. In: LNCS 294, pp 227-236
5. Chang HJ, Hou TY, Hsu A, Chang SK (1995) The management and application of tele-action objects. ACM Multimedia Systems J. 3: 204-216
6. Chang SK (1995) Towards a theory of active index. J. Visual Languages and Computing 6:101-118
7. Chang SK (1996) Extending visual languages for multimedia. IEEE Multimedia 3: 18-26
8. Chang SK, Tortora G, Yu B, Guercio A (1987) Icon purity: toward a formal theory of icons. International Journal of Pattern Recognition and Artificial Intelligence 1: 377-392
9. Costagliola G, Tortora G, Orefice S, De Lucia A (1995) Automatic generation of visual programming environments. Computer 28: 56-66
10. Costagliola G (1997) Formal methods for visual editors and compilers. Technical Report, Dipartimento di Informatica ed Applicazioni, University of Salerno
11. Ferrucci F, Tortora G, Tucci M, Vitiello G (1994) A predictive parser for visual languages specified by relational grammars. Proc. IEEE Symp. on Visual Languages, pp. 245-252
12. Ferrucci F, Pacini G, Satta G, Sessa MI, Tortora G, Tucci M, Vitiello G (1996) Symbol relation grammars: a formalism for graphical languages. Information and Computation 131:1-46
13. Golin EJ, Reiss (1990) SP the specification of visual language syntax. J. Visual Languages and Computing 1:141-157
14. Grosky W, Jain R, Mehrotra R, eds. (1997) The handbook of multimedia information management. Prentice-Hall PTR
15. Guha A, Pavan A, Liu JCL, Roberts BA (1995) Controlling the process with distributed multimedia. IEEE Multimedia 2:20-29
16. Hirakawa M Call for papers first international workshop on multimedia software engineering. URL: <http://www.huis.hiroshima-u.ac.jp/~hirakawa/MSE98/mse98.html>
17. Lin CC, Xiang JX, Chang SK (1996) Transformation and exchange of multimedia objects in distributed multimedia systems. ACM Multimedia Systems J 4:2-29
18. Marriot K, Meyer B (1996) Towards a hierarchy of visual languages. Proc. IEEE Symp. on Visual Languages, pp 196-203
19. Saiedian H (1996) An invitation to formal methods. Computer 29:16-17
20. Wittenburg K (1992) Earley-style parsing for relational grammars. Proceedings of IEEE Workshop on Visual Languages and Computing, pp 192-199
21. Weitzman L, Wittenburg K (1994) Automatic Presentation of multimedia documents using relational grammars. Proc. ACM Multimedia '94, pp 443-451
22. Weitzman L, Wittenburg K (1996) Grammar-based articulation for multimedia document design. ACM Multimedia Systems J 4:99-111

23. Weitzman L, Wittenburg K (1996) Relational grammars: theory and practice in a visual languages interface for process modeling. AVI '96 Int. Workshop on Theory of Visual Languages. URL: <http://www.cs.monash.edu.au/~berndm/TVL96/tvl96-home.html>
24. Wong STC, Huang HK (1997) Networked multimedia for medical imaging. IEEE Multimedia 4:24-35
25. Woolf BP, Hall W (1995) Multimedia pedagogues: interactive systems for teaching and learning. Computer 28:74-82