Chapter 9

Systems: The Design of Multimedia Languages

1. INTRODUCTION

The inherent complexity and size of many multimedia applications requires the introduction of proper software engineering techniques, languages, and tools for mastering the specification, the development process, and the dynamics characterizing their presentation. In Chapter 4, we described how visual languages can be extended in order to capture the dynamic behavior of multimedia objects [Chang96a].

The extended visual languages are called multidimensional languages and are still based on the concept of generalized icons [Chang91]. The user can access and animate multimedia information by composing multidimensional sentences, that is, by combining generalized icons according to some spatial and/or temporal relations. The extended visual languages can be used for the development of teleaction objects (TAOs) [ChangH95b], multimedia objects that automatically respond to events, and are particularly suitable for modeling multimedia presentations.

In this chapter we discuss the design of multidimensional languages for specific application domains. The chapter is structured as follows. In Sections 2 and 3 we describe in order the TAO model and the extended generalized icons. In Section 4 we review the concepts underlying a visual language design methodology. The new methodology is presented in Section 5, whereas an example is described in Section 6. Finally we discuss the methodology in Section 7.

2. THE TAO MODEL FOR MULTIMEDIA

TAOs are multimedia objects capable of automatically reacting to events and messages. The structure of the multimedia objects is represented through an hypergraph G, whereas the event driven dynamic structure is represented through a knowledge structure K called Active Index [Chang96a]. G is a graph G(N, L), where N is the set of nodes and L is the set of links. A node can represent a media type or even a TAO itself. Links can be of the following types: attachment, annotation, location, and synchronization. An example of TAO hypergraph is given in Figure 1.

The physical appearance of a TAO is described by a multidimensional sentence, which is a spatial/temporal composition of generalized icons [Chang91], [Chang96a]. A multidimensional language is a set of multidimensional sentences. The syntactic structure underlying a multidimensional sentence controls its dynamic multimedia presentation.

3. GENERALIZED ICONS AND ICON OPERATORS

Generalized icons are dual objects $x = (x_m, x_p)$, where x_m is the meaning and x_p is the physical appearance.

In visual languages the physical appearance x_p is an icon image. In multidimensional languages the concept of generalized icon has been extended to represent all the different types of media [Arndt97], [Chang96a]. The following types of generalized icons have been defined:

- Icon: (x_m, x_i) where x_i is an image
- Earcon: (x_m, x_e) where x_e is sound
- Micon: (x_m, x_s) where x_s is a sequence of icon images (motion icon)
- Ticon: (x_m, x_t) where x_t is text (ticon can be regarded as a subtype of 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 composite icon or multimedia sentence.

Also icon operators are dual objects $op = (op_m, op_i)$, The physical part (op_i) combines the physical parts of generalized icons, whereas the logical part (op_m) combines their meanings. Multidimensional sentences are constructed by combining generalized icons through earcon operators such as fade in or fade out, micon operators such as zoom in or zoom out, ticon operators such as text merge or text collate, and temporal operators [Allen91].

In TAOs generalized icons are represented by nodes in the hypergraph, whereas operators are represented by links. As an example, let us consider the multimedia presentation Salerno Multimediale, a CD-ROM describing the city of Salerno. The presentation begins by displaying a cover image. After the user touches the screen, a background sound is played and animation starts. The latter is composed of a background image with a rotating label "Salerno Multimediale" on it. After few seconds the rotating label fades out and a falling curtain starts covering the background image. The animation yields another background image with a menu overlaid on it. The TAO hypergraph for this portion of the CD-ROM is shown in Figure 1.

4. A METHODOLOGY FOR VISUAL LANGUAGE DESIGN

Based on the concept of generalized icons, we have developed a methodology for the design of iconic languages [Chang94b], a subclass of visual languages. Successively, the methodology has been extended to accomplish the design of general visual languages [Poles98] and temporal visual languages [Chang97].

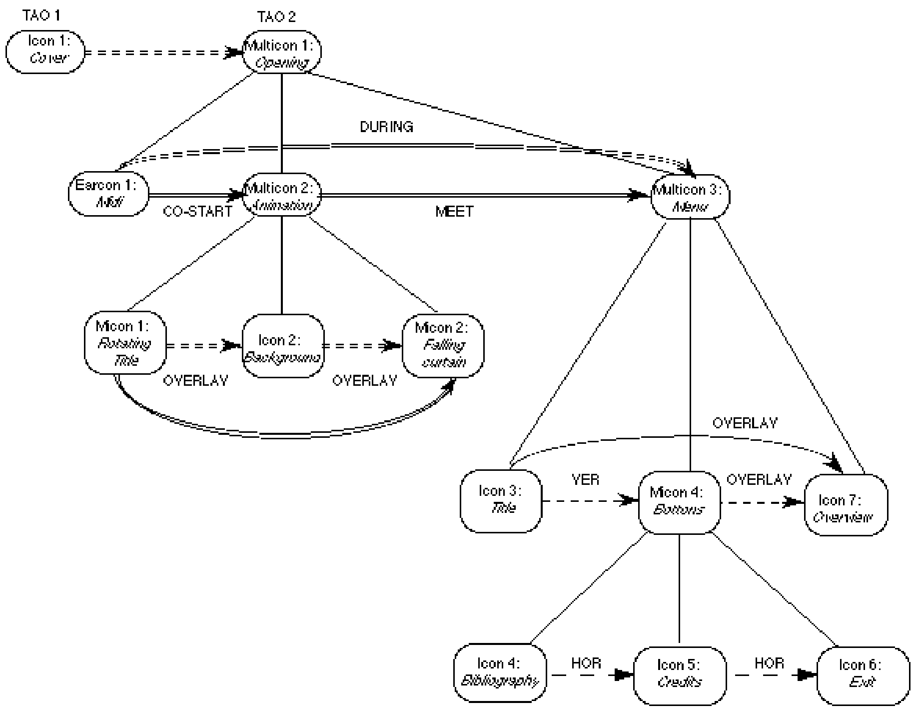

Figure 1. An example of the hypergraph structure.

The design problem for visual languages is to encode the elements of an application domain through visual sentences semantically close to them, according to a certain metaphor. Let K be the set of domain elements to be visually encoded, the phases to be executed in our design methodology are outlined in Figure 2.

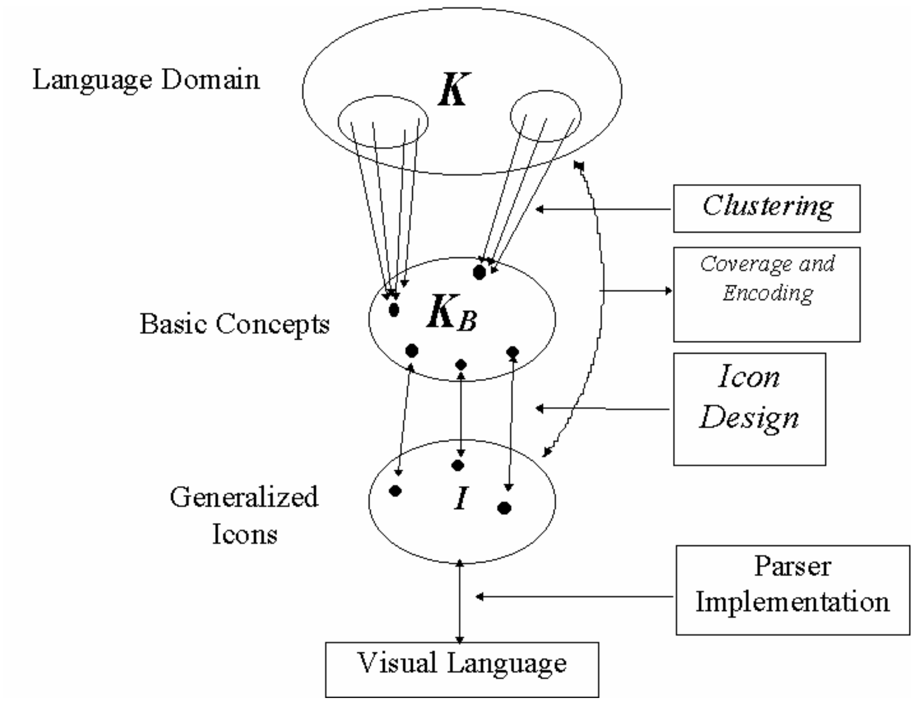

Figure 2. The methodology for visual language design.

Initially, we need to build the domain K and a reduced knowledge base to better characterize its elements. Then, we cluster K according to an adaptation of the K-B-means algorithm to obtain a reduced set KB of clusters representing basic concepts of the application domain. Successively, the phase of icon design is performed to sketch a visual representation for the elements of KB. For each word or concept $w_i \in$ KB we must sketch a visual representation for w_i to derive a new generalized icon $x = (x_m , x_i)$ such that xm includes w_i. At the end we obtain a set I of basic generalized icons. In the next step we should combine the icons in I through the operators of the icon algebra [Chang91] to form visual sentences, for the sake of visually encoding the whole set K. At this point, we should encode each element w i from the language domain through a visual sentence made of icons and operators. During the Coverage and Encoding phase we construct a visual sentence $S = (S_m , S_i)$, run inferences to derive its meaning part S_m from the meaning parts of its component icons and icon operators, and then use S_i to encode the domain element w_i similar in meaning to S_m .

The final set of visual sentences form the language icon dictionary. Each record of this dictionary contains a domain element, the visual sentence encoding it, and a formal rationale explaining the association between the word and the visual sentence. The final language is tested through special tools to verify usability for the intended users. Finally, we need to construct a visual grammar in order to generate a parser [Costa97a].

5. THE EXTENDED METHODOLOGY FOR MULTIDIMENSIONAL LANGUAGES

We have extended our design methodology for visual languages to allow the design of multidimensional languages. The design problem for multidimensional languages is to derive a multimedia representation for the elements of an application domain. In general, this process includes content selection, media allocation, and media realization [Weitz94]. We see this process as the derivation of a certain number of TAOs representing the elements of the application domain in multimedia presentations. The association between these elements and the TAO is also dynamically ruled by the knowledge structure associated to the TAO. Thus, we have defined a design process to derive the multidimensional language for expressing the TAOs for a given domain.

5.1 Domain and Knowledge Construction

In this phase we have to build the multidimensional language domain K. As opposed to visual language design here the language domain includes more types of elements, such as images, sounds, etc. The frame structure for the knowledge base includes some of the attributes used for the design of visual languages, such as the attributes sound, time, location, color, and shape. Some other attributes depend upon the application domain and are used to express content. Their values include not only text but also image and sound. As a consequence, these attributes can also have a special index to be used for similarity matching. In fact, the similarity function to be used for multidimensional languages needs to find similarity in sound or image, for which it can use well known indexing and approximate matching techniques developed for multimedia databases.

5.2 Modeling and Clustering

In this phase we first structure the domain elements by using object oriented modeling techniques and then we perform Clustering by using the class diagram and a special distance function. The distance function still compares attributes to determine the similarity, but it will be using more sophisticated and approximate matching techniques because of the presence of complex types of data. For example, a text mentioning the painting of Leonardo Monnalisa should be considered close and therefore clustered together with a figure showing the image of MonnaLisa, and with all the images having similar visual characteristics, such as colors, shapes, etc.

5.3 Generalized Icon Design

The input to this phase is the class diagram, the object diagrams and the clusters determined in the previous phase. We need to translate their information in terms of generalized icons. Thus, we first construct the physical appearance of generalized icons and then their logical part. We can decide to provide both a visual and a sound representation for a textual information according to a certain metaphor.

After have sketched generalized icons for elementary information we apply icon operators to compose multidimensional sentences. We exploit the relationships of the class diagram and the clusters to understand the appropriate operators to apply.

5.4 Approximate coverage and TAO generation

For each element of the domain we compute a set of multidimensional sentences and a score indicating the type of similarity with a rationale associated. This information will be used for TAO construction.

After have constructed a set of multidimensional sentences covering the language domain, we can build a visual grammar and semantic routines to produce the TAOs covering the domain elements. The parsing of the sentences will control the multimedia presentation.

6. CASE STUDY

In this section we show an example on the use of our methodology for the development of a multidimensional language to transform lectures into multimedia presentation formats. We started from a domain language made of textual lectures with transparencies comprising text, figures, tables, and movies on specific subjects. The goal was to derive the multidimensional sentences expressing the TAOs for the multimedia presentation of the lectures. An abstract class diagram for this example is shown in Figure 3.

Let us consider a transparency from a medical lecture on meniscal surgeries as shown in Figure 4. The text item Umbrella handle has associated the following frame in the knowledge base:

SLOT: VALUE
NAME: Umbrella handle
SHAPE: Sketch(Umbrella handle)
LOCATION: Middle of Knee
IMAGE: Overlay (This.Shape, Knee CT Scan)
TIME: Before(Before(Co start(This.text, This.Image), "Circle shape"), Movie1)

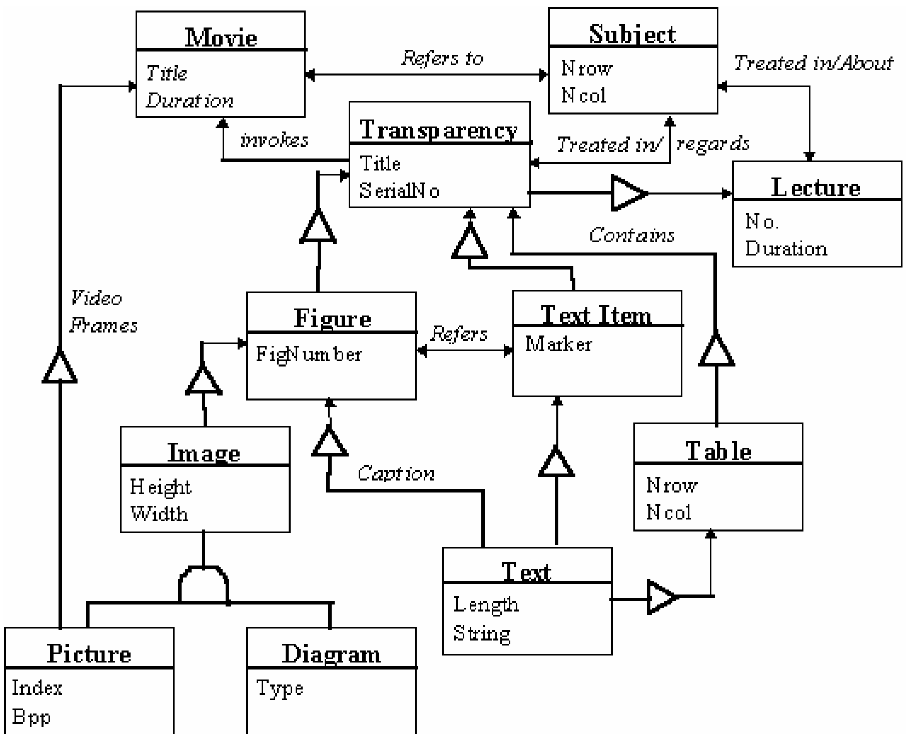

Figure 3. The class diagram for the Lecture example.

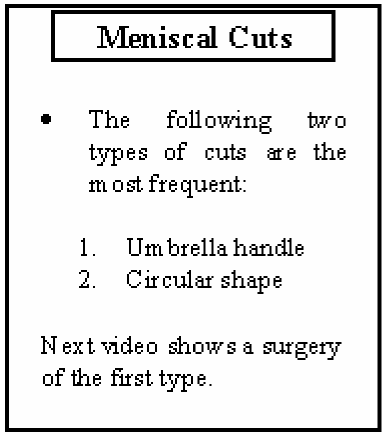

Figure 4. A transparency on Meniscal Surgeries.

In the object diagram there will be an instance Meniscal Cuts of the class Transparency, which is connected to the instance Meniscal Surgery of the class Subject through the relationship Regards, to the instance Movie1 of the class Movie through the relationship Invokes, and is composed of three instances of the class Text Item.

The Image attribute is a query to an image database [Nappi98]. We run such queries on the medical images of the language domain by using the system FIRST [Nappi98] during the clustering phase. In this way we could cluster images with similar meniscal anomalies together. During the Generalized Icon Design we produced a vicon Movie1 and a ticon for each of the three text items in the transparency. Then, the *part-of* relationship between the transparency and its three text items suggested the introduction of a multicon for the whole transparency Meniscal Cuts and the application of the attach operator for each of the three text items. We applied the spatial operator ver to combine the ticons. The Invokes relationship and the TIME attribute from the frame associated to the transparency suggested the application of the temporal operator before to combine the multicon with the vicon. The SHAPE attribute of the the textual items "Umbrella handle" and "Circular shape" suggested the sketch of two icons each depicting one of the two shapes. The sketch queries contained in the IMAGE attribute caused the linking during the clustering phase to examples of CT scan images reporting similar knee anomalies. We could then decide to enrich the presentation by combining the two ticons with either the two icons or even with some of the CT scan images (represented as icons) resulting from the queries. A further decision was to be made on the spatial and temporal operators to use for combining ticons and icons. For example, each ticon could be combined with the associated icon by using the spatial operator overlay and the temporal operators co start, co end.

The entry Meniscal Cuts in the language icon dictionary will have associated several candidate multidimensional sentences and the rationale for each of them.

An example of a rationale follows:

[Multicon[Meniscal_Cuts] *attach*
[[ticon[umbrella] *overlay* + *co_start* + *co_end* icon[umbrella_cut]]
vertical [ticon[circular] *overlay* + *co_start* + *co_end* icon[circular_cut]]]
before Vicon[Movie1]

We can easily generate the associated TAO from this rationale.

7. DISCUSSION

We have presented a methodology for the design of multidimensional languages. The methodology serves as a prescriptive model for designing multidimensional sentences to be used for visually specifying the structure of Teleaction Objects. We need to further experiment the proposed methodology on a broader class of multimedia applications. Moreover, we need to refine logical icon operators to increase the generation of knowledge for the active index of TAOs.