Chapter 3

Syntax: Visual Languages

As mentioned in Chapter 1, a multimedia application is constructed from a collection of multimedia objects. The primitive objects are media objects of the same media type. The complex multimedia objects are composed from these primitive objects and in general are of mixed media types. The syntax of a multidimensional language ML describes how the complex multimedia objects are constructed from the other multimedia objects. Spatial and temporal composition rules must be taken into consideration.

Before we consider such a multidimensional language for multimedia objects, we may begin with a visual language for visual objects. For one thing, there has been considerable research on visual languages. For another, many multimedia applications are still primarily oriented towards visual objects.

A visual language is a pictorial representation of conceptual entities and operations and is essentially a tool through which users compose iconic, or visual, sentences [Chang95b]. The icons generally refer to the physical image of an object. Compilers for visual languages must interpret visual sentences and translate them into a form that leads to the execution of the intended task [Chang90]. This process is not straightforward. The compiler cannot determine the meaning of the visual sentence simply by looking at the icons. It must also consider the context of the sentence, how the objects relate to one another. Keeping the user's intent and the machine's interpretation the same is one of the most important tasks of a visual language [Crimi90].

1. ICONS

A visual sentence is a spatial arrangement of object icons and/or operation icons that usually describes a complex conceptual entity or a sequence of operations. *Object icons* represent conceptual entities or groups of object icons that are arranged in a particular way. *Operation icons*, also called process icons, denote operations and are usually context-dependent. Figure 1 illustrates a visual sentence that consists of horizontally arranged icons, with a dialog box overlaid on it. This particular location-sensitive visual sentence changes meaning when the locations of icons change, and can be used to specify to-do items for TimeMan, a time-management personal digital assistant. Figure 2 illustrates a content-sensitive visual sentence for TimeMan. The fish in the tank are object icons, each of which represents a to-do item, and the cat is an operation icon that appears when there are too many fish in the tank (the to-do list is too long). On the other hand, Figure 4 in Chapter 4 illustrates a time-sensitive visual sentence that changes its meaning with time.

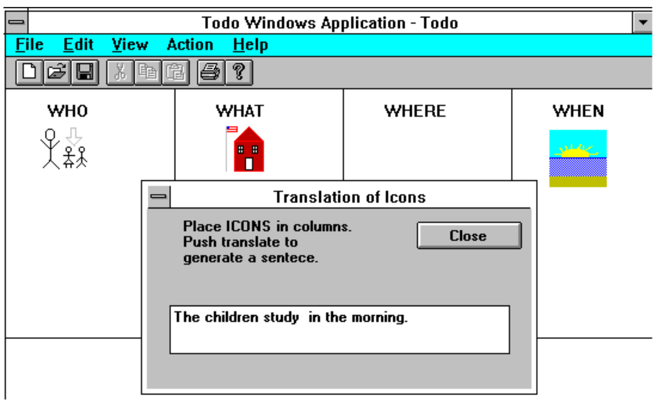

Figure 1. This figure is a location-sensitive visual sentence that shows how the placement of the “school” icon changes its meaning. Here the meaning is “The children study in the morning.” In Figure 2 of Chapter 4, the meaning is “The children drive to school in the morning.” Such visual sentences can be used to specify to-do items for the time management personal digital assistant TimeMan.

2. OPERATORS

Icons are combined using *operators.* The general form of binary operations is expressed as x_1 *op* $x_2 = x_3$, where the two icons x_1 and x_2 are combined into x_3 using operator *op*. The operator *op* = (op_m, op_p), where op_m is the logical operator, and op_p is the physical operator. Using this expanded notation, we can write (x_{m1}, x_{p1}) *op* (x_{m2}, x_{p2}) = ((x_{m1} op_m x_{m2}), (x_{p1} op_m x_{p2})). In other words, the meaning part x_{m1} and x_{m2} are combined using the logical operator op_m, and the physical part x_{p1} and x_{p2} are combined using the physical operator op_p.

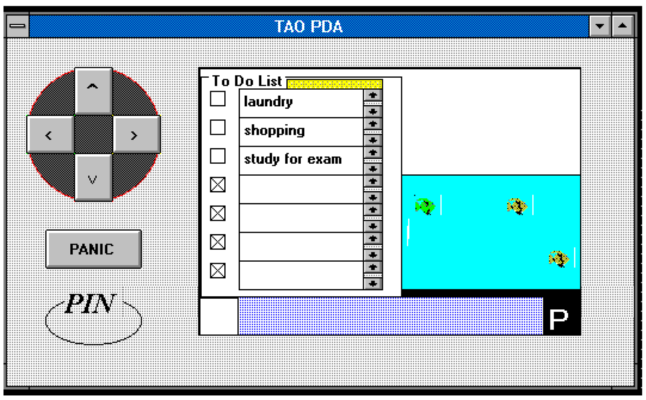

Figure 2. Content-Sensitive visual sentence shows the fish-tank-and-cat metaphor for the time management personal digital assistant TimeMan. Each fish represents a to-do item.

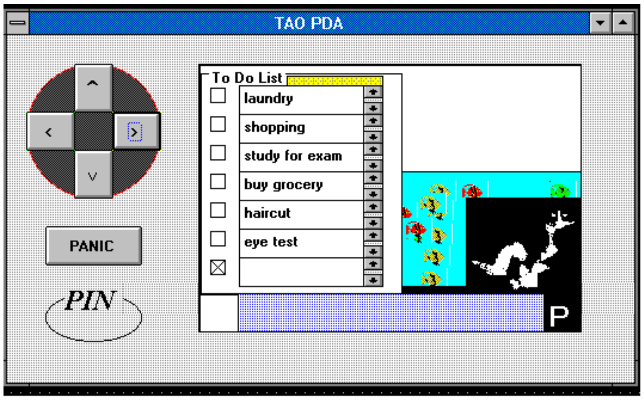

Figure 3. When the to-do list grows too long, the fish tank is overpopulated and the cat appears. The fish tank icon and cat operation icon have corresponding index cells receiving messages from these icons when they are changed by the user (see Chapter 4).

Operators can be *visible* or *invisible*. Most system-defined spatial/temporal operators are invisible, while all user-defined operators are visible for the convenience of the user. For example, excluding the dialog box, the visual sentence in Figure 1 is the horizontal combination of three icons. Therefore, it can be expressed as:

(CHILDREN *hor* SCHOOL_HOUSE) *hor* SUNRISE

where *hor* is an invisible operator denoting a horizontal combination. But if we look at Figure 2, the *cat* is a visible operator denoting a process to be applied to the fish in the fish tank. An operation icon can be regarded as a visible operator.

The four most useful domain-independent icon operators are *ver*, for vertical composition; *hor*, for horizontal composition; *ovl*, for overlay; and *con*, for connect. *ver, hor* and *ovl* are usually invisible, and *con* is usually visible as a connecting line.

The invisible icon operators are *spatial operators* and apply only to icons or ticons. The spatial composition of two icons or ticons is a *complex icon*.

3. REPRESENTING MEANING

To represent the meaning of an icon, we use either a frame or a conceptual graph, depending on the underlying semantic model of the application system being developed. Both are appropriate representations of meaning, and can be transformed into one another. For example, the SCHOOL_HOUSE icon in Figure 1 can be represented by the following frame:

```
Icon SCHOOL_HOUSE
WHO:      nil
DO:       study
WHERE:    school
WHEN:     nil
```

In other words, the SCHOOL_HOUSE icon has the meaning “study” if it is in the DO location, or the meaning “school” in the WHERE location. Its meaning is "nil" if it is in the WHO or WHEN location. An equivalent linearized conceptual graph is as follows:

```
[Icon = SCHOOL_HOUSE]
  --(sub)-->   [WHO = nil]
  --(verb)->   [DO = study]
  --(loc)-->   [WHERE = school]
  --(time)->   [WHEN = nil]
```

The meaning of a composite icon can be derived from the constituent icons, if we have the appropriate inference rules to combine the meanings of the constituent icons. We have applied conceptual dependency theory to develop inference rules to combine frames [Chang94b]. We have also adopted conceptual operators to combine conceptual graphs [Chang89]. As a simple example, the merging of the frames for the icons in the visual sentence shown in Figure 1 will yield the frame:

Visual_Sentence vs_1

```
WHO:      children
DO:       study
WHERE:    nil
WHEN:     morning
```

We can derive this frame by merging the frames of the four icons using the following rule:

The i^{th} slot gets the value of the corresponding slot of the i^{th} icon.

Thus the first slot with slot_name WHO gets the value “children” from the corresponding slot of the first icon CHILDREN, the second slot with slot_name DO gets the value "study" from the corresponding slot of the second icon SCHOOL_HOUSE, etc.