See this page in another language

Chapter 2 Visual Computing

Visual Computing

Visual computing is computing on visual objects. Some visual objects such as images are **inherently visual** in the sense that their primary representation is the visual representation. Some visual objects such as data structures are **derivatively visual** in the sense that their primary representation is not the visual representation, but can be transformed into a visual representation. Images and data structures are the two extremes. Other visual objects such as maps may fall somewhere in between the two. Visual computing often involves the transformation from one type of visual objects into another type of visual objects, or into the same type of visual objects, to accomplish certain objectives such as information reduction, object recognition and so on.

In visual computing it is important to ask the following question: who performs the visual computing? The answer to this question determines the approach to visual computing. For instance it is possible that primarily the computer performs the visual computing and the human merely observes the results. It is also possible that primarily the human performs the visual computing and the computer plays a supporting role. Often the human and the computer are both involved as equal partners in visual computing and there are visual interactions. Formal or informal visual languages are usually needed to facilitate such visual interactions. With the advances in bio-computing it is conceivable that visual computing may involve animals, robots, cyborgs and other hybrid life forms so that visual languages can be either natural or artificial.

Figure 1 illustrated the paradigm for visual computing.

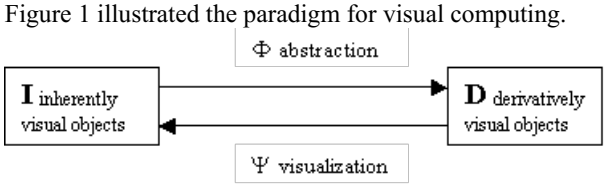


Figure 1. Transformation among visual objects.

Visual Language

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].

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(a) 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). Figure 3 illustrates a time-sensitive visual sentence that changes its meaning with time.

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_p 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. Operators can be visible or invisible. Most system-defined spatial/temporal operators are invisible, whileas all user-defined operators are visible for the convenience of the user. For example, excluding the dialog box, the visual sentence in Figure 1(a) 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; ov, 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.

Grammar

Visual languages can handle temporal as well as spatial operators. A visual language has a relational grammar, G, which a compiler uses to generate sentences:
G = (N, X, OP, s, R)
where N is the set of nonterminals, X is the set of terminals (icons), OP is the set of spatial relational operators, s is the start symbol, and R is the set of production rules whose right side must be an expression involving relational operators.

Syntax

Informally, a visual language is a set of visual sentences, each of which is the spatialcomposition of icons. Figure 1(b) without the dialog box illustrates a simple visual sentence, which describes the physical appearance of an object retrieved by BookMan. With the dialogue box, the figure becomes a multidimensional sentence used by BookMan to generate **The children drive to school in the morning**. in synthesized speech. The multidimensional sentence has the syntactic structure
(DIALOG_BOX co_start SPEECH) ver (((CHILDREN hor CAR) hor SCHOOL_HOUSE) hor SUNRISE)
Figure 3 is a hypergraph of the syntactic structure. The syntactic structure is essentially a tree, but it has additional temporal operators (such as co_start) and spatial operators (such as hor and ver) indicated by dotted lines. Some operators may have more than two operands (for example, the co_start of audio, image, and text), which is why the structure is called a hypergraph. The syntactic structure can be used to control the multimedia presentation.
To describe multidimensional languages, we can extend the X and OP elements of G: X is still the set of terminals but now includes earcons, micons, ticons, and vicons as well as icons, and the OP set now includes temporal as well as spatial relational operators.

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(a) 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(a) will yield the frame:

> Visual_Sentence vs1
> 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 ith slot gets the value of the corresponding slot of the ith 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.

Figures:

Figure 1. A visual sentence whose meaning changes when the icons change their positions and is called a location-sensitive sentence. The visual sentence (a) has the meaning **The children study in the morning**, and (b) has the meaning **The children drive to school in the morning.** Comparing the two, this example shows how the placement of the **school** icon changes the meaning. Such visual sentences can be used to specify to-do items for the time management personal digital assistant TimeMan.
Figure 1(a)

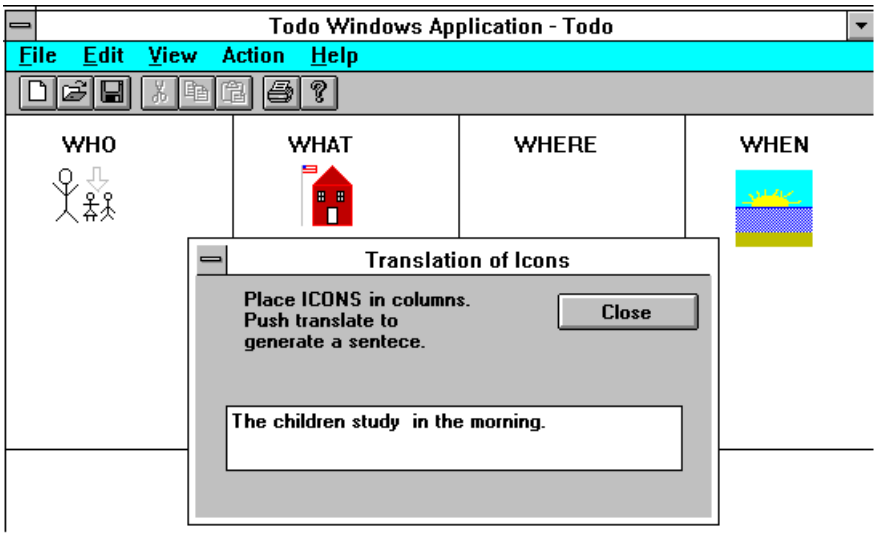


Figure 1(b)

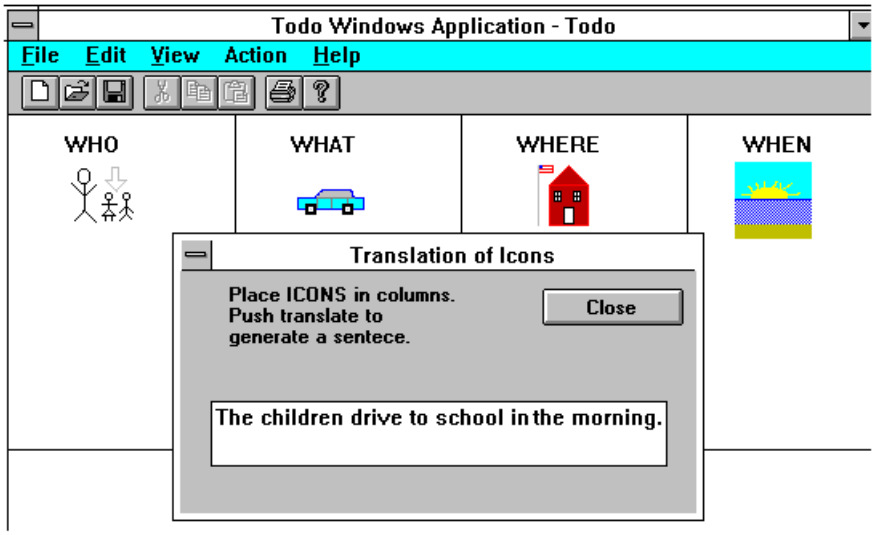


Figure 2. Content-Sensitive visual sentences (a) and (b) show the fish tank and cat metaphor for the time management personal digital assistant TimeMan. Each fish represents a to-do item. When the to-do list grows too long, the fish tank is overpopulated and the cat appears. The fish tank icon and cat operation icon have corresponding index cells receiving messages from these icons when they are changed by the user.
Figure 2(a)

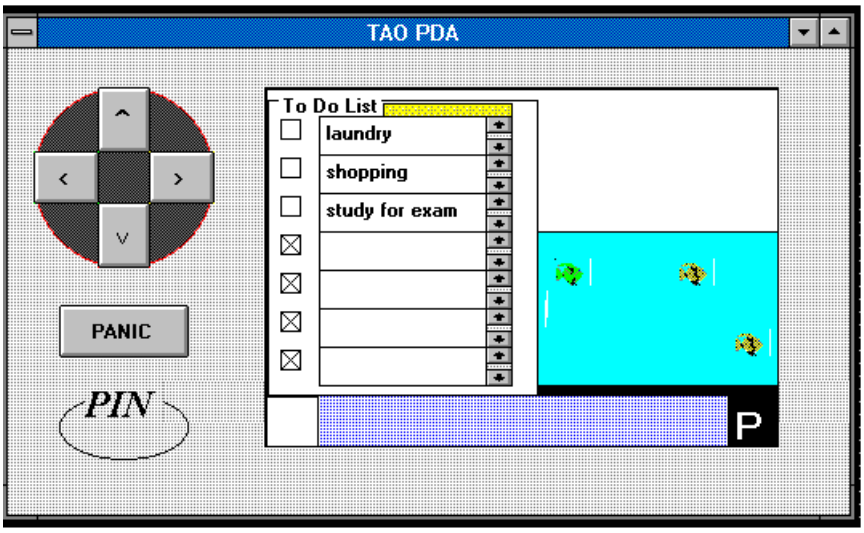


Figure 2(b)

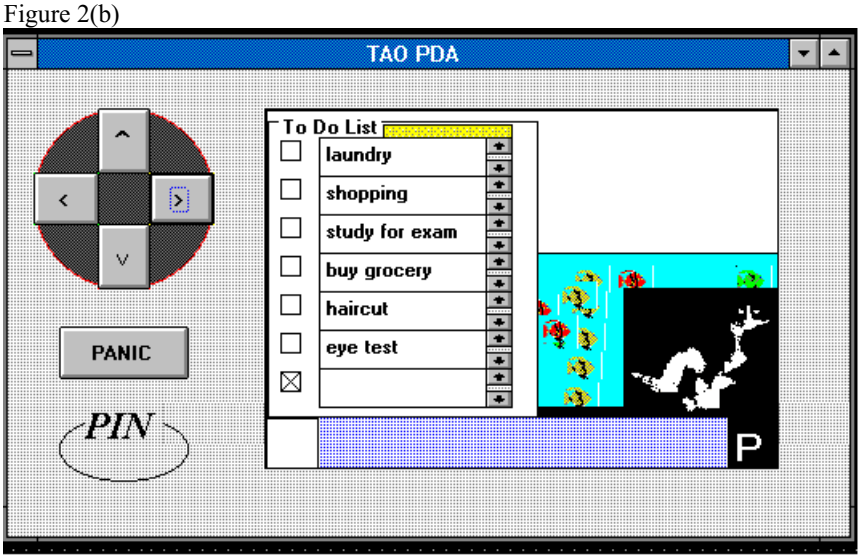
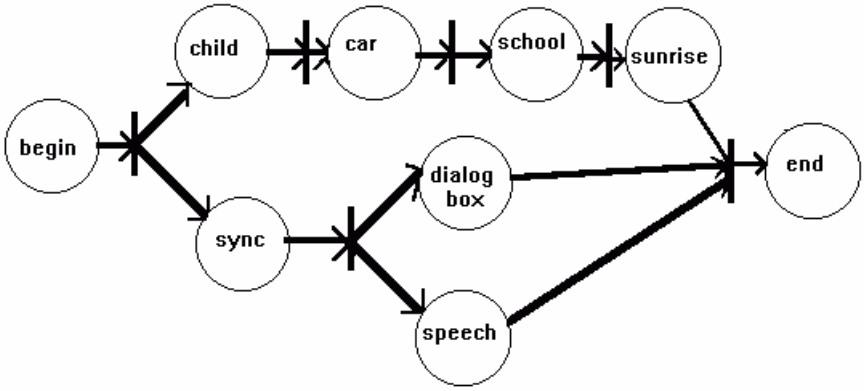


Figure 3. A time-sensitive visual sentence for the Petri net controlling the presentation of the visual sentence shown in Figure 1(b).



BIBLIOGRAPHY

[ALLEN83] Allen, J. F., "Maintaining Knowledge about Temporal Intervals," Communications of the ACM, vol. 26, no. 11, pp. 832-843, November 1983.
[CHHC95] Chang, H., T. Hou, A. Hsu, and S. K. Chang, "Management and Applications of Tele-Action Objects," ACM Multimedia Systems Journal, vol. 3, no. 5-6, pp. 204-216, Springer Verlag, 1995.
[CHANG87] Chang, S. K., "Icon Semantics - A Formal Approach to Icon System Design," International Journal of Pattern Recognition and Artificial Intelligence, vol. 1, no. 1, pp. 103-120, 1987.
[CHANG89] Chang, S. K., M.J. Tauber, B. Yu, and J.S. Yu, "A Visual Language Compiler," IEEE Transactions on Software Engineering, vol. 5, no. 5, pp. 506-525, 1989.
[CHANG90] Chang, S. K., "A Visual Language Compiler for Information Retrieval by Visual Reasoning," IEEE Transactions on Software Engineering, pp. 1136-1149, 1990.
[CHANG94a] Chang, S. K., M. F. Costabile, and S. Levialdi, "Reality Bites - Progressive Querying and Result Visualization in Logical and VR Spaces," Proc. of IEEE Symposium on Visual Languages, pp. 100-109, St. Louis, October 1994.
[CHANG94b] Chang, S. K., S. Orefice, M. Tucci, and G. Polese, "A Methodology and Interactive Environment for Iconic Language Design," International Journal of Human- Computer Studies, vol. 41, pp. 683-716, 1994.
[CHANG95a] Chang, S. K., "Towards a Theory of Active Index," Journal of Visual Languages and Computing, vol. 6, no. 1, pp. 101-118, 1995.
[CHANG95b] Chang, S. K., G. Costagliola, G Pacini, M. Tucci, G. Tortora, B. Yu, and J. S. Yu, "Visual Language System for User Interfaces," IEEE Software, pp. 33-44, March 1995.
[CRIMI90] Crimi, C., A. Guercio, G. Pacini, G. Tortora, and M. Tucci, "Automating Visual Language Generation," IEEE Transactions on Software Engineering, vol. 16, no. 10, pp. 1122-1135, October 1990.
[KHALIFA96] Khalifa, Y., S. K. Chang, and L. Comfort, "A Prototype Spatial- Temporal Reasoning System for Emergency Management," Proc. of International Conference on Visual Information Systems VISUAL96, pp. 469-478, Melbourne, Australia, February 5-7, 1996.
[LIN96] Lin, C. C., J. X. Xiang, and S. K. Chang, "Transformation and Exchange of Multimedia Objects in Distributed Multimedia Systems," ACM Multimedia Systems Journal, vol. 4, no. 1, pp. 2-29, Springer Verlag, 1996.