FILOMAT (Niš) 8 (1994), 65–70

PURE ICONS IN ICONIC SYSTEM

T. Kitić

Abstract. *This paper turn to some theoretical considerations that are necessary for a deeper understanding of the behavior of icons in an iconic system. Only for pure icons the logical part be completely recovered from the physical part. Impure icons may cause problems when used for person-machine communication, for building iconic user interfaces and for writing formal specifications. So, this paper describes purity-preserving conditions for iconic operators.*

1. Introduction

A unifying concept for visual languages is that they all deal with different aspects of generalized icons [CHA87]. Generalized icons consist of object icons or process icons. An object icon is a dual representation of an object, written as $(x, X_m, X_i) = (X_m, X_i)$, with a logical part X_m (the meaning) and a physical part X_i (the image). Visual programming language deal with objects that have logical meaning, but no visual image.

A **process icon** [CHA87] represents an action or a computational process. Language constructs handled by a visual language are process icons, or icons with a logical and physical part that represents a computational process.

An **iconic system** [CHA87] is a structured set of related icons. The iconic system G is a **formal iconic system**, syntactic specification of the iconic system, if it is represented as a quintuple $G(VL, VP, S, x_0, R)$, where VL is a set of logical objects, VP is a set of physical objects, S is a finite, nonempty set of icon names, x_0 is an element in S, denoting the head icon name and R is a mapping from $S \mapsto 2^{VL\,S}X\,VP$, denoting icon rules. The icon rules R specify icons as the dual representation by a set of logical objects and a physical object, or formally:

$$R = \{(x, X_m, X_i) | x ::= (X_m, X_i) \in R\}$$

Given an iconic system G, we can classify all icons as **elementary, composite or structural.**

* **Elementary icon** is an icon where is $X_m \cap S = \emptyset$. In other words, $X_m \subset VL$, so that x is of the form $(\{labels\}, image)$. The labels could denote names of objects, procedures or operators, so that the elementary icon can be an object

Received 23.02.1994; Revised 05.11.1994
1991 *Mathematics Subject Classification*: 68U10

icon, a process icon or an operator icon.
There are special elementary icons. An **image icon** is one where $X_m = \emptyset$, so x is of the form $(\{\ \}, image)$. A **label icon** is one where the physical part is null, so x is of the form $(\{labels\}, e)$. A **null icon** is of the form $(\{\ \}, c)$.

* **Complex icon** is an icon where is $X_m \cap VL \neq \emptyset$. A complex icon points to other icons and define icon relations. There are the following types:

a) **composite icon**, where $X_m \cap VL \neq \emptyset$. Then, the icon x is of the form $(\{OP, y_1, \ldots, y_n\}, image)$, where y_i are subicons or logical objects, and OP is an iconic operator which operates on the subicons $y_1, \ldots, y_n$ to create a new icon. The location attributes of the subicons will determine the order in applying the iconic operator.

b) **structural icon**, is if $X_m \cap VL = \emptyset$. The icon is of the form $(\{y_1, \ldots, y_n\}, image)$. In other words, x is related to $y_1, \ldots, y_n$, but the mechanism for composing x from $y_1, \ldots, y_n$ is unspecified.

Iconic operators are special icons which carry certain meanings based on action to be performed. The basic actions performed by iconic operators include: constructing the image of a resultant icon, synthesizing descriptions, determine the type label of resultant icon, determine the common attributes of icons and optionally performing evaluation.

Iconic operators are used to represent certain relationships between icons. These relation ship could be physical relationships or logical relationships or both. Iconic operators operate on generalized icons and change either the logical part or the physical part of an icon, or both. So, iconic operator $OP(2)$ has two parts:

$$OP = (OP_m, OP_i) \tag{1.1}$$

or OP_m for the logical operator and OP_i for the physical operator.

A **binary iconic operator** OP[SC90] has two arguments, X and Y:

$$OP(X,Y) = [OP_m(X,Y),\ OP_i(X,Y)] \tag{1.2}$$

and when there is no mutual dependency, that is when OP_m does not depend on X_i or Y_i and OP_i does not depend on X_m or Y_m, we can write

$$OP(X,Y) = [OP_m(X_m,Y_m),\ OP_i(X_i,Y_i)] \tag{1.3}$$

A **unary iconic operator** OP[SC90] has only one argument X:

$$OP(X) = [OP_m(X),\ OP_i(X)] \tag{1.4}$$

and when there is no mutual dependency, we can write

$$OP(X) = [OP_m(X_m),\ OP_i(X_i)] \tag{1.5}$$

Definition 1. The **materialization function** MAT[SC90] is a partial function from $WL = VL \cup S \mapsto 2^{VP}$, defined as:

$$MAT(X_m) = \begin{cases} \{X_i; (X_m, X_i) \in R\} \\ \text{undefined for other } X_m \end{cases} \tag{1.6}$$

Definition 2. The **dematerialization function** DMA[SC90] is a partial function from $VP \mapsto WL$, defined as:

$$DMA(X_i) = \begin{cases} \{X_m; (X_m, X_i) \in R\} \\ \text{undefined for other } X_i \end{cases} \tag{1.7}$$

Definition 3. An generalized icon (X_m, X_i) **is pure**, if

$$\begin{cases} MAT(X_m) & = \{X_i\} \\ DMA(X_i) & = \{X_m\} \end{cases} \tag{1.8}$$

For pure icon only, the logical part can be completely recovered from the physical part. This is possible, only when $MAT(X_m)$ and $DMA(X_i)$ are both singletons.

In general, $MAT(X_m)$ may yield a set of icon images. For example, MAT(Mona-Lisa) may be the original drawing of Mona-Lisa, or a sketch of Mona-Lisa. $DMA(X_i)$ may also yield a set of meanings. Such impure icons may cause problems if used for person-machine communication, specially writing formal specifications of user interfaces. So, this paper try to give some theoretical considerations that are necessary for a deeper understanding of the behavior of icons in an iconic system and help to many writers of formal specifications to understand the notation of data abstractions in an iconic system and writing the specification of physical part of an icon and the specification of logical part of an icon.

2. Determination of icons purity

Because, we can classify all icons in a few categories, the way of determination their purity is different.

2.1. Elementary icons and their purity. For elementary icons, it is easy to determine directly their purity from MAT and DMA. Here is an example.

For some image database, the formal iconic system is G, $(\{c_1, c_2\}, \{p_0, p_1, p_2, p_3, p_4, p_5\}, \{x_0, x_1, x_2, x_3, x_4, x_5\}, x_0, R_1)$, where the icon rules R_1 are

$$\begin{cases} (x_0, \{x_1, x_2\}, p_0 \\ (x_1, \{x_3, x_4\}, p_1 \\ (x_2, \{\ \ \}, p_2 \\ (x_3, \{c_1\}, p_3 \\ (x_4, \{c_2\}, p_4 \\ (x_5, \{c_3\}, p_5 \end{cases} \tag{1.9}$$

The MAT and DMA functions are

$$(1.10)\qquad \begin{cases} MAT(\{x_1,x_2\})=\{p_0\} \\ MAT(\{x_3,x_4\})=\{p_1\} \\ MAT(\{c_1\})=\{p_3\} \\ MAT(\{c_2\})=\{p_4,p_5\} \end{cases} \qquad \begin{cases} DMA(p_0)=\{\{x_1,x_2\}\} \\ DMA(p_1)=\{\{x_3,x_4\}\} \\ DMA(p_2)=\{\ \ \} \\ DMA(p_3)=\{\{c_1\}\} \\ DMA(p_4)=\{\{c_2\}\} \\ DMA(p_5)=\{\{c_2\}\} \end{cases}$$

Icon x_3 is pure, while icons x_4 and x_5 are not. Icon x_2 is an image icon. It is pure in this example, but if the iconic system has another image icon, x_2 will no longer be pure. Since image icons are physical images without any label, we usually do not think of them as pure icons.

If we use the partial functions MAT and DMA, then the structural icons x_0 and x_1 are also pure. Again we usually do not think of structural icons as pure icons, because they normally represent **iconic relations**.

2.2. Complex icon.

2.2.1. Complex icons and their purity.

Definition 4. A **composite icon** [SC90] $(x, X_m = \{OP, y_1, \ldots, y_n\}, X_i)$ is composed from subicons $y_1, \ldots, y_n$ as :

$$\begin{cases} X_m = OP_m(y_1, \ldots, y_n) \\ X_i = OP_i(y_1, \ldots, y_n) \end{cases}$$

where $OP(y_1, \ldots, y_n)$ is an n-array iconic operator.

As define before in 1.1, 1.4 and 1.5, we can state

(C-1) $\quad MAT[OP_m(y_1, \ldots, y_n)] = \{OP_i(MAT(Y_{m_1}), \ldots, MAT(Y_{m_n}))\}$

(C-2) $\quad DMA[OP_i(y_1, \ldots, y_n)] = \{OP_m(DMA(Y_{i_1}), \ldots, DMA(Y_{i_n}))\}$

or condition (C-1) says that $MAT(X_m)$, or the materialization of X_m, is equal to the application of the operator OP_i on the individual materialization of the subicons $Y_{m_1}, \ldots, Y_{m_n}$. Condition (C-2) can be interpreted similarly.

The consequence of these purity-preserving conditions is state in the next theorem.

Theorem 1. *If $y_1, \ldots, y_n$ are pure icons and the purity-preserving conditions hold, then the composite icon x is also pure.*

Proof. We need to show $MAT(X_m) = \{X_i\}$ and $DMA(X_i) = \{X_m\}$.

$$\begin{aligned} MAT(X_m) &= MAT[OP_m(y_1, \ldots, y_n)] = \\ &= OP_i\{(MAT[(Y_{m_1}), \ldots, MAT(Y_{m_n})])\} && \text{(because of C-1)} \\ &= \{OP_i(\{Y_{i_1}\}, \ldots, \{Y_{i_n}\}\} && \text{(because } Y_i \text{ are pure)} \\ &= \{X_i\} \\ DMA(X_i) &= DMA[OP_i(y_1, \ldots, y_n)] = \\ &= \{OP_m(DMA[(Y_{i_1}), \ldots, DMA(Y_{i_n})])\} && \text{(because of C-2)} \end{aligned}$$

$$= \{OP_m(\{Y_{m_1}\}, \ldots, \{Y_{m_n}\}\} \qquad \text{(because } Y_m \text{ are pure)}$$
$$= \{X_m\}$$

2.2.2. Structural icons and their purity. Structural icons can be regarded as composite icons with an implicit operator STC. Therefore, if $(x, X_m = \{y_1, \ldots, y_n\}, X_i)$ is a structural icon, we have

$$X_m = STC_m(y_1, \ldots, y_n) = \{y_1, \ldots, y_n\}$$
$$X_i = STC_i(y_1, \ldots, y_n) = STC_i(Y_{i_1}, \ldots, Y_{i_n}).$$

It is very important to remember that, we usually do not think of structural icons as pure icons, because they normally represent iconic relations.

3. Conclusion

For composite icon (x, X_m, X_i), the X_m part is formally denoted by $\{OP, y_1, \ldots, y_n\}$. By that, it means $OP_m(y_1, \ldots, y_n)$, but it will formally write $\{OP, y_1, \ldots, y_n\}$. If $OP_m(y_1, \ldots, y_n)$, really generates a new meaning, then the composite icon x becomes once more an elementary icon: that is, $(x, X_m, X_i) = [x, OP_m(y_1, \ldots, y_n), OP_i(y_1, \ldots, y_n)]$. So, if the purity hold, this newly composed icon is also a pure icon. The implications of the purity-preserving conditions are the following:

If we find the meanings of all subimages, then we can combine them to find the meaning of the whole image. (C-3)

If we find the images of all the partial meanings, then we can combine them to find the image of the whole meaning. (C-4)

It should be noted that the purity-preserving conditions also imply

$$\begin{cases} OP_m(y_1, \ldots, y_n) & = OP_m(y_{m_1}, \ldots, y_{m_n}) \\ OP_i(y_1, \ldots, y_n) & = OP_i(y_{i_1}, \ldots, y_{i_n}) \end{cases}$$

or in other words,

$$OP(y_1, \ldots, y_n) = [OP_m(y_{m_1}, \ldots, y_{m_n}), OP_i(y_{i_1}, \ldots, y_{i_n})]$$

So, this defined the purity of individual icons. This should help us to determine unambiguous formal specification for formal iconic system. It is very important characteristic of good formal specification. The materialization (MAT) and dematerization (DMA) operators allow representation conversion among different types of icons and construction an image processing language with icon-assisted navigation [CHA85], [CH85].

REFERENCES

[CHA85] S.K. Chang, E. Jungert, S. Levialdi, G.Tortora, and T.Ichikawa, *An Image Processing Language with Icon-Assisted Navigation*, IEEE Transaction on Software Engineering, Avgust 1985, 811-819.

[CH85] U.L. Chi, *Formal Specification of User Interfaces: A Comparison and Evalutation of Four Axiomatic Approcoaches*, IEEE Transactions on Software Eng., SE-11, 8(1985), 671-685.

[CHA87] S.K. Chang, *Visual languages: A Tutorial and Survey*, IEEE Software, January 1987, 29-39.

[SC90] S.H. Chang, *"Principles of Visual Programming Systems"*, New Jersy, Prentice Hall, Englewood Cliffs, 1990.

MIN Institut, Višegradska 33
18 000 Niš, YUGOSLAVIA