Part III
Chapter 18

Transformations of Object Diagrams

18.1 Introduction

This chapter introduces some of the most common transformations of object diagrams. This is done in an informal and pragmatic manner; the proof of the correctness of the transformations is intended for Part IX, Formal Basis.

Many transformations of object diagrams depend on a knowledge of the system that is being modelled. A knowledge of the relationships between the objects is also required to determine whether transformations are allowed. In Chapter 6, many relationships between objects were described. Most of these relationships were introduced to illustrate the links between the diagrams and mathematical theory. Several mathematical manipulations were seen to result in valid transformations of object diagrams.

This chapter will concentrate on the composition and decomposition of objects and connections between objects. It must be seen as a list of some of the most common transformations, that when sensibly applied to a model of a system, may lead to new insights, and aid the understanding of the relationship between systems and system components. There is no claim as to the completeness of the list presented here.

18.2 Reference section

In the following manual pages, the following manipulations are illustrated:

Parallel connections
Decomposition
Refinement
Composition
Serial composition
Pseudo independence
Abstraction with epochs

NAME

Parallel connections

PURPOSE

To simplify or complicate the connectivity pattern in an obje diagram.

SYNOPSIS

Several parallel connections between two objects can be shown a one provided the alphabet carried by the single connection can b expressed as the composite of the alphabets on the constituen connections. If an alphabet can be expressed as a composit alphabet, then the connection can be decomposed into a number (parallel connections.

CANONICAL FORM

The canonical forms are shown in Figure 18.1. Several connection between $Object_1$ and $Object_2$ may be combined into a singl connection, if the new connection carries a composite alphabet. Th reverse transformation is also allowed.

Figure 18.1 Parallel connections and composite alphabets

SPECIFICATION

The justification for the transformations can be found in a simpl renaming exercise. The composite alphabet J×K is renamed I Each element in P is a pair (j,k), and the cardinality of P and of J×I must be the same.

REALIZABILITY ISSUES

SEE ALSO

COMPOSITE ALPHABET
COMPOSITE INTERACTION

NAME

Decomposition

PURPOSE

To decompose an object into a combination of objects.

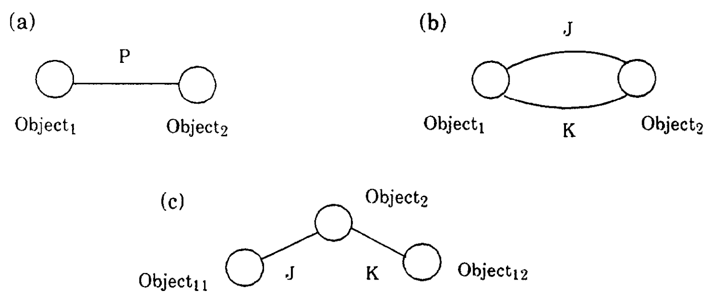
SYNOPSIS

A pair of simple objects linked by a single connection can be represented by parallel connections. The description of the new objects is derived from the description of the original objects by re-expressing the constraints imposed by the objects in terms of the combined alphabets of the parallel connection. The decomposition can be carried out when the constraints of one of the original objects can be represented by a pair of independent constraints.

CANONICAL FORM

There are three stages in a decomposition. A pair of objects linked by a connection, shown in figure 18.2(a) can be represented by a pair of objects with a pair of parallel connections as shown in figure 18.2(b). If the relation maintained by one of the objects can be represented as a cross-product of two independent constraints then that object can be decomposed into two independent objects to form a serial connection of three objects as shown in figure 18.2(c).

Figure 18.2 Re-arrangement of objects and connections



SPECIFICATION

The first step in the decomposition is to form a composite alphabet $J \times K$ whose members correspond to members of the alphabet P. The composite alphabet must have at least as many members as the original alphabet. If the alphabet P has n_p members, alphabet J, n members and alphabet K, n_k members then

$$n_j \, . \, n_k \geq n_p$$

There are two ways of relating the two alphabets. Firstly al members of the composite alphabet can map to a member of the original alphabet

M: J × K to P

Editorial: the nature of these functions needs further investigation.

where M is a surjective function. If the composite alphabet is larger than the original alphabet then it implies that there are alternative ways of representing members of the alphabet P. This is referred to as redundant mapping.

Alternatively the alphabet P can be related to the composite alphabet by a injective function

N: P to J × K

in this case some members of the composite alphabet cannot be said to represent any of the members of the original alphabet. Some of the members of the composite alphabet will be unused. This is referred to as supernumary mapping.

The second step is to recouch the constraints imposed by the objects in terms of the composite alphabet. In a simple case, suppose $Object_1$ maintains constraint P_1 and $Object_2$ maintains the constraint P_2 which is re-expressed as R_{jk2} and that the constraint P_1 can be written as a composition of *independent* constraints

$$J_1 \times K_1$$

then $Object_1$ can be decomposed into two objects: $Object_{11}$ and $Object_{12}$. $Object_{11}$ maintains the constraint J_1 while $Object_{12}$ maintains the constraint K_1 and $Object_2$ in Figure 18.2(c) maintains the relation R_{jk2}.

For the other two types of alphabet mapping, the expression of P_1 and P_2 are not as simple.With redundant mapping the duplicate symbols are incorporated in a constraint if its duplicate partner is included.

Suppose $Object_1$ maintains constraint P_1 and $Object_2$ maintains the constraint P_2 then the re-expressed constraint for $Object_1$ is a relation, R_{jk1}, given by

$\{ (j, k) \mid M(j, k) \in P_1 \}$

and R_{jk2} is the relation

$\{ (j, k) \mid M(j, k) \in P_2 \}$

With supernumary mapping the relations must exclude the possibility of a supernumary symbol occurring and the relation R_{jk1} is given by

$\{ (j, k) \mid \exists\, p . (Np = (j, k) \wedge p \in P_1) \}$

and R_{jk2} is the relation

$\{ (j, k) \mid \exists\, p . (Np = (j, k) \wedge p \in P_2) \}$

REALIZABILITY ISSUES

SEE ALSO

PARALLEL CONNECTIONS
COMPOSITE ALPHABET
COMPOSITE INTERACTION
INDEPENDENCE RELATIONSHIP
COMPLEX OBJECTS

FUTURE DIRECTIONS

NAME

Refinement

PURPOSE

To refine the specification of one of a pair of objects, withou changing the interactions between the two.

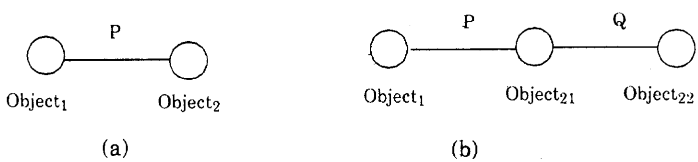
SYNOPSIS

Either one of a pair of simple objects linked by a single connectio can be refined into a complex and a simple object linked by a singl connection.

CANONICAL FORM

The canonical forms are shown in Figure 18.3. A pair of interactin objects, $Object_1$ and $Object_2$ can be represented as three objects connected in a serial fashion. To achieve this $Object_2$ has bee refined into two parts: $Object_{21}$ and $Object_{22}$.

Figure 18.3 Refinement of $Object_2$



SPECIFICATION

The refinement of $Object_2$ must not lead to changes in the potentia interactions as seen from $Object_1$. Suppose $Object_1$ imposes th constraint P_1 and $Object_2$ imposes the constraint P_2.

By refining $Object_2$ a new relationship is introduced: R_{pq}, impose by $Object_{21}$. If $Object_{22}$ imposes the constraint Q_2 on it interactions with $Object_{21}$ then Q_2 is constrained by

$\{ q \mid \exists p . ((p, q) \in R_{pq} \wedge p \in P_2) \}$

or R_{pq} is given by

$P_2 \times Q_2$

REALIZABILITY ISSUES

SEE ALSO

COMPLEX OBJECT

FUTURE DIRECTIONS

NAME

Composition

PURPOSE

To allow the composition of objects and connections.

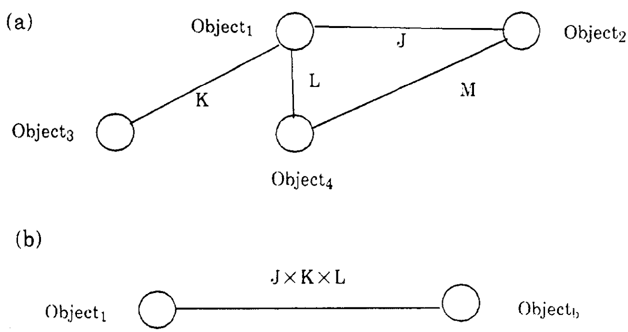
SYNOPSIS

A number of connected objects can be represented by a pair o interacting objects linked by a single connection carrying ε composite alphabet.

CANONICAL FORM

The canonical form is shown in two system representations. In one illustrated in Figure 18.4(a) several objects are connected togethe: in a particular fashion. The canonical form of the compositior consists of two objects, with a single connection, as illustrated ir Figure 18.4(b).

Figure 18.4 A system of interconnected objects (a), and the canonical form of the composition (b)

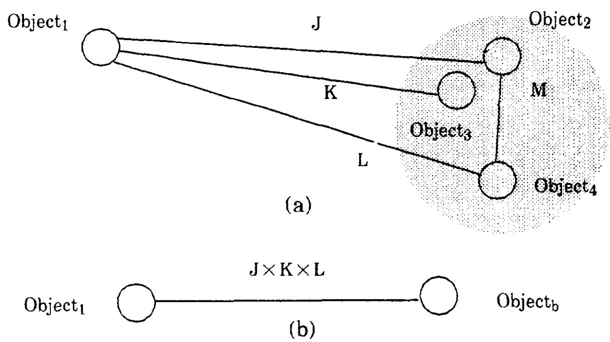


SPECIFICATION

A system containing several objects can be reduced to a canonica form by agglomerating objects. Figure 18.5(a) shows the diagran of the system, shown in Figure 18.4(a), rearranged with $Object_1$ t(the left and all the other objects to the right. The objects ar(interconnected in the same way as before. In Figure 18.5(b) $Object_2$ $Object_3$ and $Object_4$ have been combined into $Object_b$ and th(

connections and the alphabets of interaction have been merged. A single composite alphabet based on the set of symbols $J \times K \times L$ represents the alphabet of the parallel combination of connections. The role of $Object_1$ is to maintain a relation between the parts of the composite alphabet. A similar rearrangement can be done for each of the original objects to demonstrate that each object maintains a relation between the symbols at its interfaces.

Figure 18.5 Re-arranging several objects into the canonical form



REALIZABILITY ISSUES

SEE ALSO

COMPOSITE INTERACTION
COMPLEX OBJECT

FUTURE DIRECTIONS

Extension of the specification.

NAME

Serial composition

PURPOSE

To allow the composition and decomposition of objects and connections.

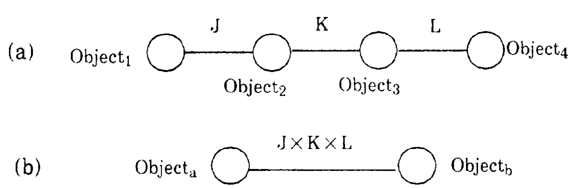
SYNOPSIS

A number of objects connected in a serial string can be represented by pair of interacting objects linked by a single connection carrying a composite alphabet. Under certain circumstances a pair of objects linked by a connection carrying a composite alphabet can be represented by a serial string of objects.

CANONICAL FORM

Serial composition is represented by a number of objects connected together to form a string as shown in figure 18.6(a). By manipulating the specifications of the objects the serial string can be replaced by a pair of objects as shown in figure 18.6(b).

Figure 18.6 Re-arranging several objects into the canonical form



SPECIFICATION

The objects in figure 18.6(a) can be rearranged in a number of ways without changing the description of the system. For example in figure 18.7(a) the objects have been grouped into two clusters. The figure shows, that with a particular arrangement of objects, the connections can be considered to be a parallel set and they can be represented by a single connection carrying the composite alphabet J×K×L. Figure 18.7(b) shows the composite connection and the two clusters of objects represented by two individual objects which each maintain the constraints originally attributed to pairs of objects.

Figure 18.7 "Folding over" the diagram before composition

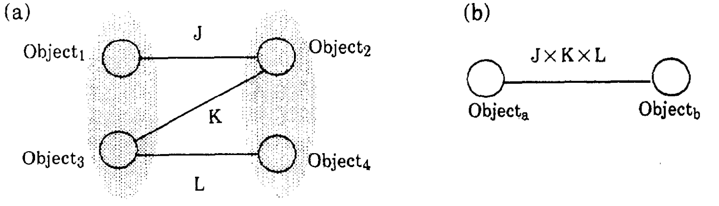
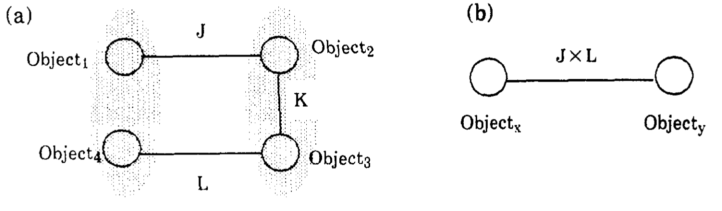


Figure 18.8(a) shows one possible alternative arrangement of the serial string of objects given in figure 18.6(a). This alternative arrangement leads to a representation with two new objects linked by a connection carrying the alphabet J×L.

Figure 18.8 Alternative arrangement of objects before composition



REALIZABILITY ISSUES

SEE ALSO

COMPOSITE INTERACTION
COMPLEX OBJECT

FUTURE DIRECTIONS

Extension of the specification.

NAME

Pseudo independence

PURPOSE

To simplify the structure of object diagrams.

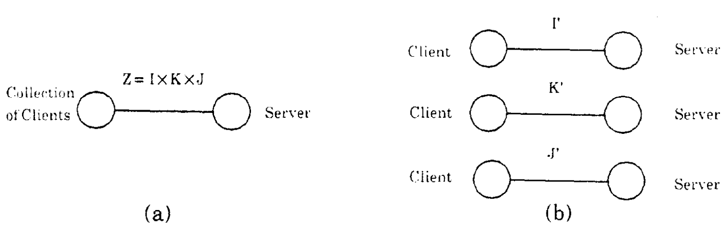
SYNOPSIS

One assumption that is often made is that a service that is shared between clients maintains independence between the interactions with the various clients. This assumption may lead to a simplified structural view of the system under consideration.

CANONICAL FORM

The canonical form shows two stages of object diagrams. In Figure 18.9(a), a collection of clients are connected to a server object. The clients are independent of one another. The assumption that the server maintains independence relationships between the interactions with the various clients, may lead to a decomposition, as shown in Figure 18.9(b), where several independent systems can be analyzed instead of one complicated system. Often the interaction alphabets in the three separate systems need to be extended to justify the assumptions about independence.

Figure 18.9 Interactions can be decomposed into separate parts.



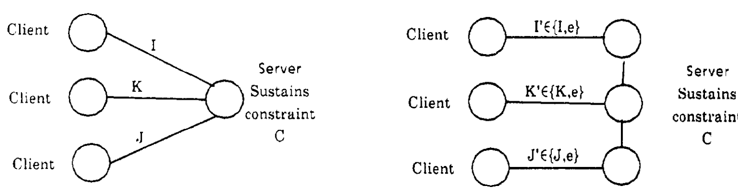
SPECIFICATION

If independence can be assured, then the alphabets in Figure 18.9(b) can be defined as:

$I' = I$
$J' = J$
$K' = K$

Sometimes independence cannot be assured and the interaction can be split but the activities of the separate parts of the interaction place constraints upon one another. These constraints have to be maintained by at least one object and possibly both. It is frequently assumed that clients are independent and a server provides a number of services that in most circumstances can be treated as being independent. Occasionally this assumption proves to be invalid because, for example, the server provides a set of services that rely on one set of computational resources. Engaging in one interaction places a constraints on other possible interactions. Figure 18.10 shows a system with three clients. The server cannot be decomposed into three separate units because it imposes a relation, C, between the three potential interactions.

Figure 18.10 The server constrains independence



Clients are built assuming that the server provides a well defined interaction and from the perspective of the clients the interactions are independent.

To create an impression of independence, the alphabet between each client and the server is extended to include exceptions. When, because of the constraint, C, the server is unable to engage in a specified interaction, it engages in an exceptional interaction. The specification of the server is well-defined but not fully determined. The system can then be treated as though it has independent clients and servers.

NAME

Abstraction with epochs

PURPOSE

To decompose a system into a number of subsystems, that can be separately analyzed.

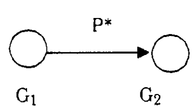
SYNOPSIS

A part of a system that is connected with another part exclusively through directed interfaces, can be decomposed into two parts. Each part can be analyzed whilst the other part is completely or partly ignored. The part of the system with the directed interface cannot influence the other part. This latter part can thus be analyzed separately, and its influence on the part with the directed interfaces can be regarded as a constant constraint for the duration of some epoch.

CANONICAL FORM

The canonical form is illustrated in Figure 18.11. A group of objects labelled G_1 is connected to another group of objects G_2 through a directed interface. An indefinite interaction is specified between the two subsystems. In some circumstances the connection between the groups can be broken and the groups can be analyzed separately.

Figure 18.11 Canonical form

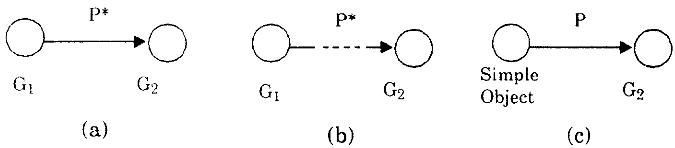


SPECIFICATION

In Figure 18.12(a) the objects in group G_2 cannot impose any constraints on the objects in group G_1. The system of objects in group G_1 can thus be separately analyzed, in isolation from G_2. For the purpose of analyzing G_1 the connection between the groups can thus be broken, as in Figure 18.12(b), and G_2 can be ignored.

The objects in group G_2 are constrained by G_1. For those periods or epochs, that G_1 imposes an unchanging constraint on G_2, G_1 can be

Figure 18.12 Separate analysis of two connected sub-systems



replaced by a simple object, that only maintains the constraint in question. This is illustrated in Figure 18.12(c).

REALIZABILITY ISSUES

SEE ALSO

EPOCH
CAUSAL OBJECT
INDEFINITE INTERACTION

FUTURE DIRECTIONS

18.2 Examples

Editorial:
Examples are to be supplied.