# Part III
# Chapter 19

# Miscellaneous Concepts

## 19.1 Introduction

The previous chapters of part III all presented a set of concepts and definitions that were somehow related to one another. By contrast, this chapter defines a number of concepts that are not necessarily related. The purpose of this chapter is to act as a first receptacle for templates that have recently been developed or for those that appear out of place with any of the existing sets.

This chapter then is best regarded as a convenience for change control.

## 19.2 Reference section

The manual pages that follow contain the description of:

Library
Directory
Epoch
System

## AME

Library

## URPOSE

To provide a way of packaging a collection of objects, whilst retaining their identity

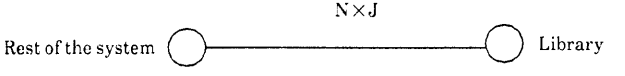
## YNOPSIS

A library is an object that consists of a collection of objects and a catalogue or directory with information about these objects. The directory helps in the identification of the objects in the library. The objects themselves may be unnamed objects. The library may then give the impression that the objects are named. Interactions with the library consist of two parts. One part identifies the object in the library (by its catalogued name), the other part relates to the interaction between the object in the library and the user of the library.

## ANONICAL FORM

A library is represented by an object in object diagrams. The canonical form shown in Figure 3.1 consists of two objects, one is the library, the other is the rest of the system and contains the objects that can interact with the library. Each interaction between the rest of the system and the library consist of two parts.

**Figure 3.1 A library**




## PECIFICATION

## EALIZABILITY ISSUES

## EE ALSO

COLLECTION OF OBJECTS
NAME INTERPRETER

NAMING MODEL
DIRECTORY

## FUTURE DIRECTIONS

The specification for the library will be based on the specifications for name interpreter, naming model and collection of objects.

## AME

Directory

## URPOSE

To identify an object in response to a name

## YNOPSIS

When a directory is presented with a name, it responds with another name. This second name is then interpreted as the identity of some object. The second name is an alternative name.

A directory can be seen as a collection of associations. The associations are between the user of the directory and other objects about which information is held in the directory.

## ANONICAL FORM

## PECIFICATION

## EALIZABILITY ISSUES

## EE ALSO

NAMING MODEL
NAME INTERPRETER
LIBRARY

## UTURE DIRECTIONS

The directory is a denotation. In terms of chapter 11 (Names) a directory is a naming model. It is not clear whether the duplicate name policies and the absent object policies, that form part of a naming model, are incorporated in the directory.

## NAME

Epoch

## PURPOSE

To distinguish one interval of time from another.

## SYNOPSIS

An epoch is a period of time. Often epochs are used to simplify object diagrams. All those things that do not change or that are insignificant over a particular interval of time may be simplified and sometimes removed from the model. It is as if for the epoch under consideration, a new set of assumptions are valid. The analysis of a system may thus be greatly simplified, but will also have a reduced validity.

## CANONICAL FORM

Epochs do not have a canonical form. A system may be abstracted into a simpler form in a particular epoch by isolating a piece of the system that has only directed interfaces.

## SPECIFICATION

## REALIZABILITY ISSUES

## SEE ALSO

EPOCHS AND TRANSPARENCY
ABSTRACTION WITH EPOCHS

## FUTURE DIRECTIONS

It is not clear whether a separate entry for epoch is sustainable. Epochs are always used in combination with another concept, never on their own.

## NAME

System

## PURPOSE

To describe a system

## SYNOPSIS

A system may be represented in two ways:

- In terms of its constituent objects and the configuration.
- In terms of an alphabet and a relation.

In the latter form, a system can be regarded as an object with an interface. As described in Chapter 2, such an object can be defined by two alphabets. The first alphabet identifies the set of possible interactions that the system is willing to deal with and the second is the constraint that the system places upon this set. The alphabet and the constraint can be arbitrarily decomposed into a composite alphabet and a relation that defines the constraints on the composite alphabet.

Only certain decompositions will lead to a set of realizable objects and connections.

## CANONICAL FORM

The graphical representation for a system is a collection of objects, connected to one another in some pattern.

## SPECIFICATION

A system that is defined in terms of an alphabet A and a relation R has, following the convention introduced in chapter 2, a specification of the form:

{A, R}

The alphabet and the relation may be decomposed to exhibit some of the structure of the system. For instance, if the alphabet and relation could be decomposed as

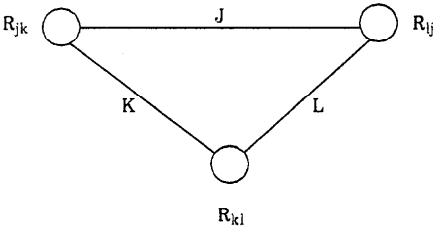
$A = J \times K \times L$

and

$R = R_{jk} \times R_{kl} \times R_{lj}$

Then the system defined by {A, R} could be represented graphically as illustrated in Figure 3.2.

**Figure 3.2 A system of interconnected objects**




## REALIZABILITY ISSUES

Only particular decompositions of a system lead to realizable objects and connections. The designer of a system is responsible for conducting the decomposition of a system specification in such a way that realizable components result.

## SEE ALSO

SYSTEM AS A STRUCTURE

## FUTURE DIRECTIONS

The transformations from the graphical to the algebraic notation will be defined explicitly, but probably not in this entry.

## 19.3 Examples