**Data models**

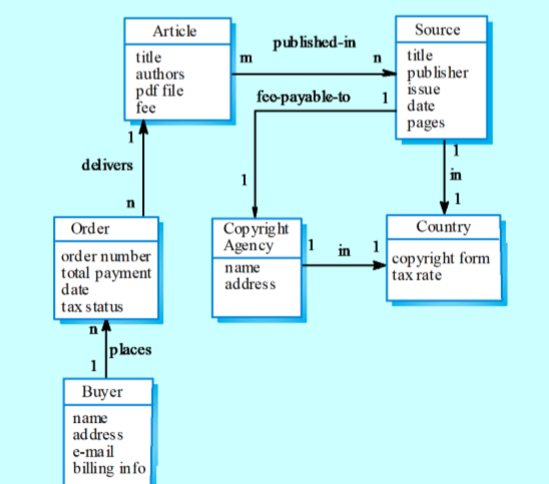
used to describe the logical structure of data processed by the system

an entity-relation-attribute model sets out the entities in the system,the relationships between these entities and the entity attributes

widely used in database design.can readily be implemented using relational databases

no specific notation provided in the uml but objects and associations can be used

eg.library semantic model



**data dictionaries**

data dictionaries are lists of all of the names used in the system models.descriptions of the entities,relationships and attributes are also included

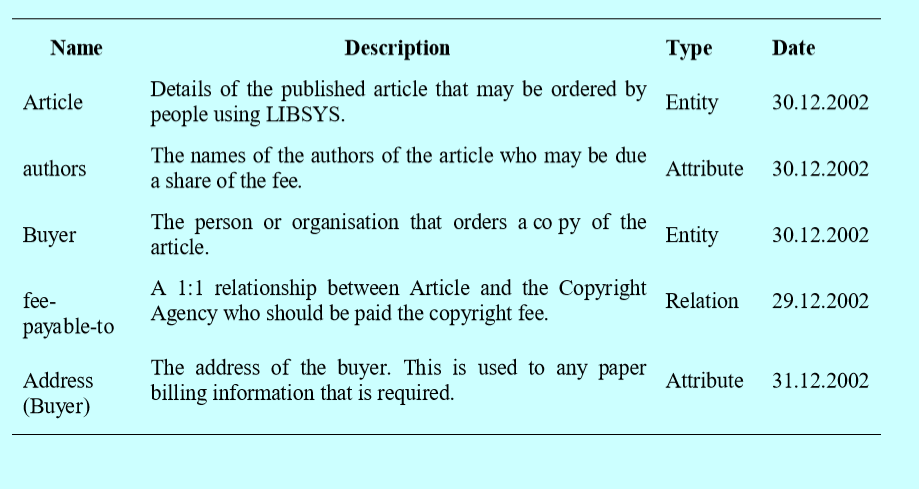
advantages

support name managements and avoid duplication

store of organisational knowledge linking analysis,design and implementation

many case workbenches support data dictionaires

eg.



**object models**

object models describe the system in terms of object classes and their associations

an object class is an abstraction over a set of objects with common attributes and the services(operations)provided by each object

various object models may be produced

inheritance models

aggregation models

interaction models

natural ways of reflecting the real-world entities manipulated by the system

more abstract entities are more difficult to model using this approach

object class identification is recognised as a difficult process requiring a deep understanding of the application domain

object classes reflecting domain entities are reusable across systems

**inheritance models**

organise the domain object classes into a hierarchy

classes at the top of the hierarchy reflect the common features of all classes.

object classes inherit their attributes and services from one or more super-classes.these may then be specialised as necessary.

class hierarchy design can be a difficult process if duplication in different branches is to be avoided.

**object models and the uml**

the uml is a standard representation devised by the developers of a widely used object-oriented analysis and design methods

it has become an effective standard for object-oriented modelling.

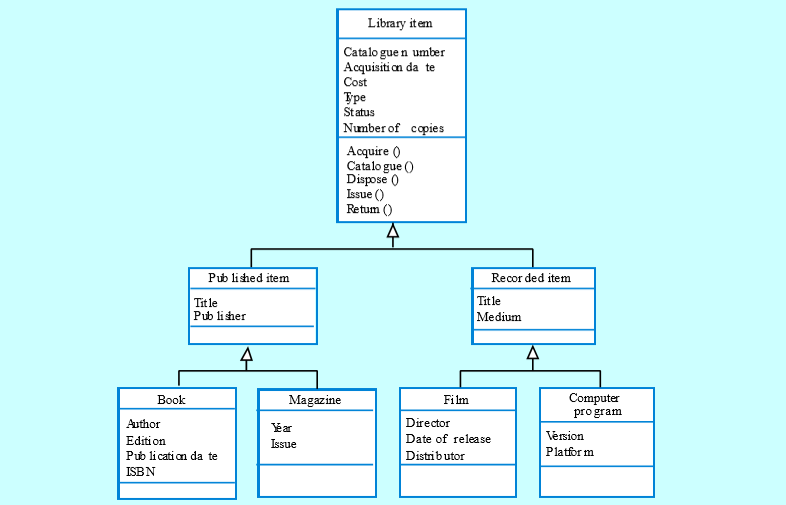
notation

object classes are rectangles with the name at the top,attributes in the middle section and operations in the bottom section;

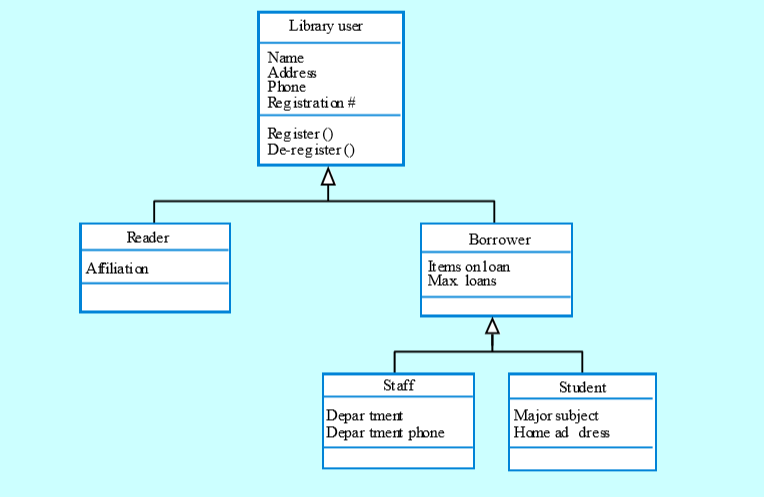
relationships between object classes(known as associations)are shown as lines linking objects

inheritance is referred to as generalisation and is shown 'upwards' rather than 'downwards' in a hierarchy

eg.library class hierarchy



eg.user class hierarchy



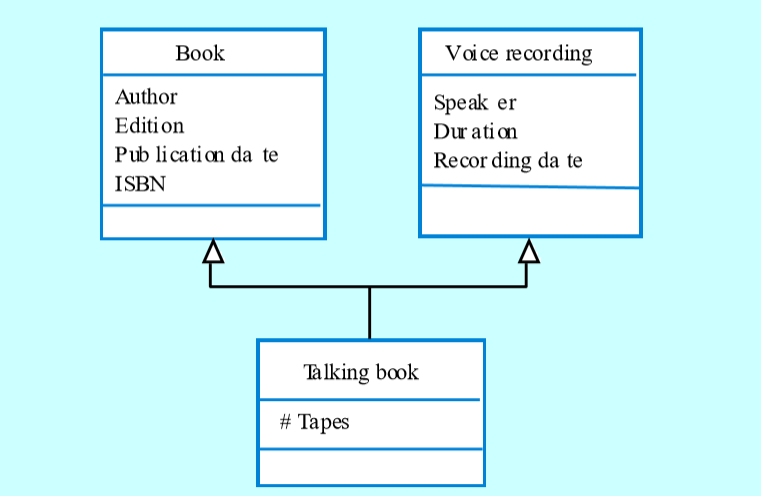
**multiple inheritance**

rather than inheriting the attibutes and services from a single parent class,a system which supports multiple inheritance allow object classes to inherit from several super-classes

this can lead to semantic conflicts where attributes/services with the same name in different super-classes have different semantics

multiple inheritance makes class hierarchy reorganisation more complex

eg.

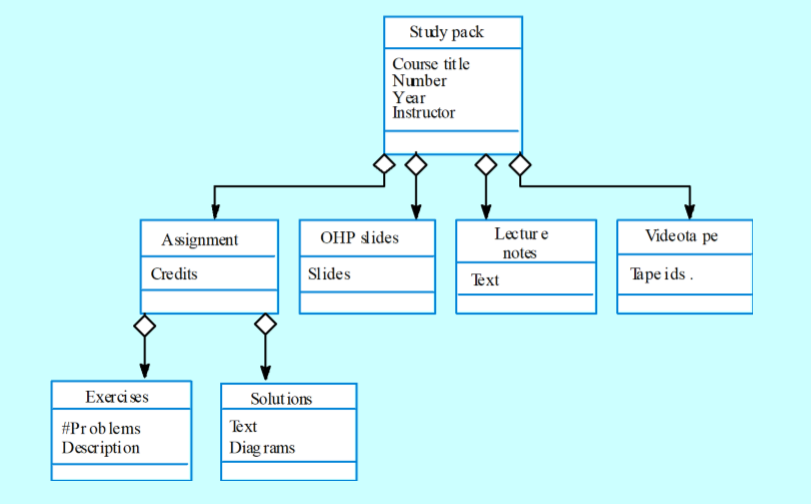


**Object aggregation**

an aggregation model shows how classes that are collections are composed of other classes

aggragation models are similar to the part-of relationship in semantic data models

eg.



**object behaviour modelling**

a behavioural models hows the interactions between objects to produce some particular system behaviour that is specified as a use-case

sequence diagrams (or collaboration diagrams) in the uml are used to model interaction between objects

**structured methods**

structured methods incorporate system modelling as an inherent part of the method.

methods defines a set of models, a process for deriving these models and rules ans guidelines that should apply to the models.

CASE tools support system modelling as part of a structured method.

method weaknesses

they do not model non-fucntional system requirements

they do not usually include information about whether a method is appropriate for a given problem.

the may produce too much documentation

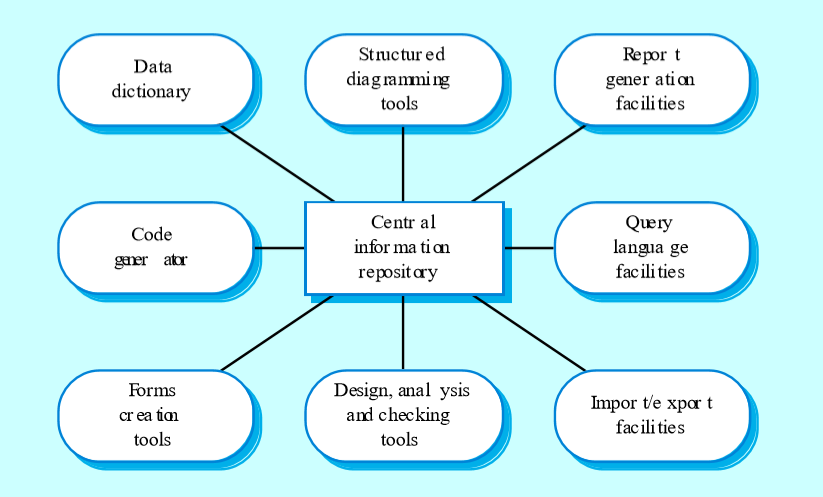
the system models are sometimes too detailed and difficult for users to understand.

CASE workbenches

a coherent set of tools that is designed to support related software process activities such as analysis,design or testing.

analysis and design workbenches support system modelling during both requirements engineering and system design.

these workbenches may support a specific design method or may provide support for a creating several different types of system model.



**analysis workbench components**

diagram editors

model analysis and checking tools

respository and associated query language

data dictionary

report definition and generation tools

forms definition tools

import/export translators

code generation tools