**software architecture**

the design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design.

the output of this design process is a description of the software architecture.

**architectural design**

an early stage of the system design process

represents the link between specification and design processes.

often carried out in parallel with some specification activities.

it involves identifying major system components and their communications.

**advantages of explicit architecture**

stakeholder communication

architecture may be used as a focus of discussion by system stakeholders

system analysis

means that analysis of whether the system can meet its non-functional requirements is possible

large-scale reuse

the architecture may be reusable across a range of systems.

**architecture and system characteristics**

performance: localise critical operations and minimise communications use large rather than fine-grain components

security: use a layered architecture with critical assets in the inner layers

safety: localise safety-critical features in a small number of sub-systems.

avaliability: include redundant components and mechanisms for saault tolerance

maintainability:use fine-grain,replaceable components

**architecture conflicts**

using large-grain components improves performance but reduces maintainability

introducing redundant data improves availability but makes security more difficult

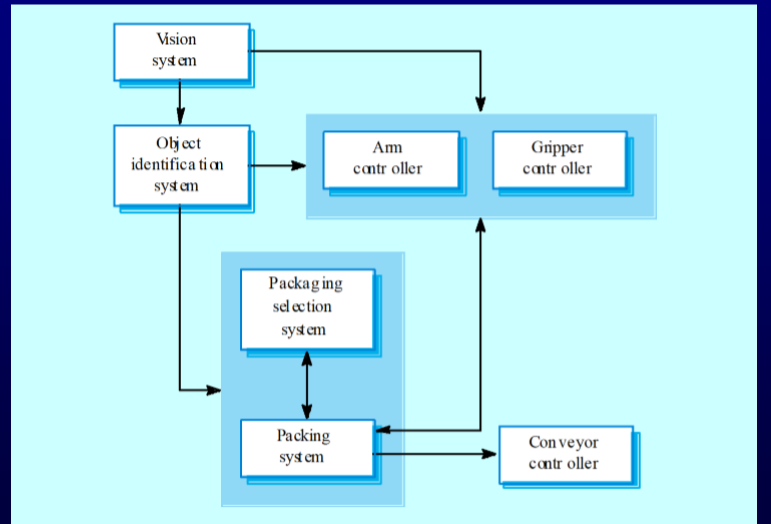
localising safety-related features usually means more communication so degraded performance

**system structuring**

concerned with decomposing the system into interacting sub-systems

the architectural design is normally expressed as a block diagram presenting an overview of the system structure

more specific models showing how sub-systems share data,are distributed and interface with each other may also be developed.



**box and line diagrams**

very abstract - they do not show the nature of component relationships nor the externally visible properties of the sub-systems.

however,useful for communication with stake holders and for project planning

**architectural design decisions**

architectural design is a creative process so the process differs depending on the type of system being developed.

however , a number of common decisions span all design processes.

**architectural design decisions**

is there a generic application architecture that can be used?

how will the system be distributed

what architectural styles are appropriate

what approach will be used to structure the system

how will the system be decomposed into modules

what control strategy should be used

how will the architectural design be evaluated

how should the architecture be documented.

**architecture reuse**

systems in the same domain often have similar architectures that reflect domain concepts

application product lines are built around a core architecture with wariants that satisfy particular customer requirements.

application architectures are covered in chapeter 13 and product lines in chapter 18

**architecturla styles**

the architectural model of a system may conform to a generic architectural model of style

an awareness of these styles can simplify the problem of defining system architectures

however,most large syatems are heterogeneous and do not follow a single architectural style.

**architectural models**

used to document an architectural design

static structural model that shows the major system components

dynamic process model that shows the process structure of the system

interface model that defines sub-system interfaces

relationships model such as a data-flow model that shows sub-system relationships

distribution model that shows how sub-systems are distributed across computers

**system organisation**

reflects the basic strategy that is used to structure a system

three organsational styles are widely used:

a shared date repository style

a shaed services and servers style

an abstract machine or layered style

**the respository model**

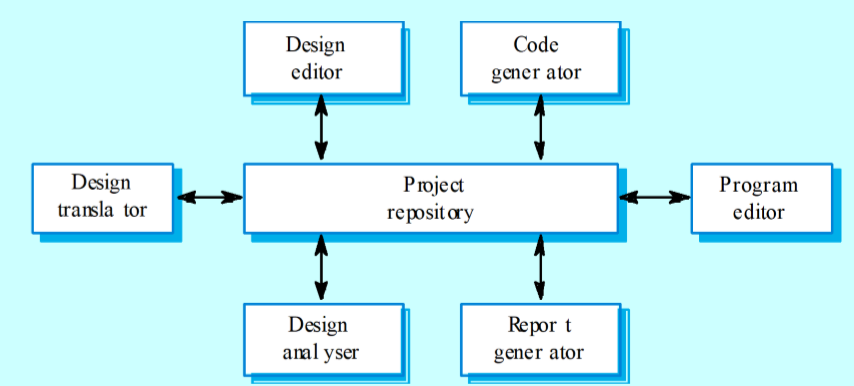
sub-systems must exchange data.this may be done in two ways:

shared data is held in a central database or respository and may be accessed by all sub-systems.

each sub-system maintains its own database and passes data explicitly to other sub-systems.

when large amounts of data are to be shared,the repository model sharing is most commonly used.

CASE toolset architecture



**repository model characteristics**

advantages

efficient way to shar large amounts of data

sub-systems need not be concerned with how data is produced centralised management eg. backup security etc.

sharing model is published as the repository schema

disadvantages

sub-systems must agree on a repository data model.inevitably a compromise

data evolution is difficult and expensive

no scope for specific management policies

difficult to distribute efficienty

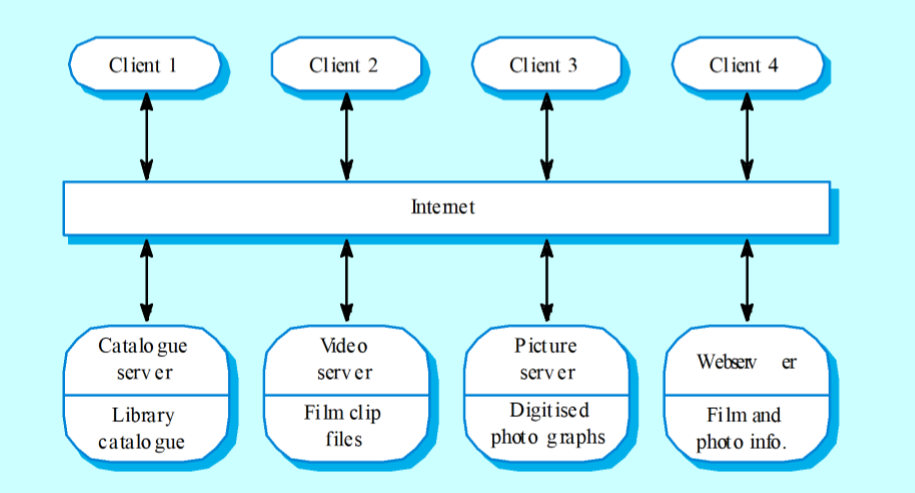
**Client-server model**

 Distributed system model which shows how data and processing is distributed across a range of components.

 Set of stand-alone servers which provide specific services such as printing, data management, etc.

 Set of clients which call on these services.

 Network which allows clients to access servers.



**Client-server characteristics**

 Advantages

• Distribution of data is straightforward;

• Makes effective use of networked systems. May require cheaper hardware;

• Easy to add new servers or upgrade existing servers.

 Disadvantages

• No shared data model so sub-systems use different data organisation. Data interchange may be inefficient;

• Redundant management in each server;

• No central register of names and services - it may be hard to find out what servers and services are available.

**Abstract machine (layered) model**

 Used to model the interfacing of sub-systems.

 Organises the system into a set of layers (or abstract machines) each of which provide a set of services.

 Supports the incremental development of subsystems in different layers. When a layer interface changes, only the adjacent layer is affected.

 However, often artificial to structure systems in this way.

**Architectural models**

 Different architectural models may be produced during the design process

 Each model presents different perspectives on the architecture

**Architecture attributes**

 Performance

• Localise operations to minimise sub-system communication

 Security

• Use a layered architecture with critical assets in inner layers

 Safety

• Isolate safety-critical components

 Availability

• Include redundant components in the architecture

 Maintainability

• Use fine-grain, self-contained components