~DRAFT~

Chapter 7. Introduction to distributed systems

Table of Contents

1. Context

This chapter shows how the network technology already treated in the previous chapters of the module can be used as a basis for creating a distributed systems.

2. Introduction

A Distributed System is a system with components that although they are linked by a network are geographically scattered. Nevertheless, such a system appears to each of its users to be a single monolithic system both in form and function. The secret of making a dispersed set of components appear to be one single system is, of course, the software that is designed to hold the system together. Ensloe's model is introduced as a way of expressing the extent to which a system is actually distributed and, so, the size of the task faced by the unifying software in making it appear to be a single system.

The properties required of such a system, such as transparency and scalability are examined, as are the characteristics that are inherent in such a system, notably fault tolerance and resource sharing. Ways of implementing a distributed system so as to ensure that it has these properties are described.

A fully functional Distributed System can be seen as a collection of sharable resources, available to all the users of the system. As such it can be used by a community with access to the supporting network to meet all their information and computational needs. In particular, distributed applications can be constructed especially for their use by combining the relevant resources from those possessed by the system.

3. Objectives

At the end of this module, you should be able to:

- explain what is meant by a Distributed System;
- explore the properties required of a Distributed System and examine the ways in which these
- properties may be ensured;
- explain how a network can support not only one Distributed System but also many different Distributed Systems at the same time;
- consider the potential for business usage of a fully functional Distributed System.

4. What is a Distributed System?

In parallel with this topic, you should read G Coulouris et al, "Distributed Systems Concepts and Design", (5th edn.).

A network can provide access to each member of a scattered collection of resources and services and can link them together so that they can communicate with each other. This allows a user to make use of each resource or service, and to send an output from one to become an input to another. This is all very well, but with any complex task the user must carry out all the component sub-tasks and, even worse, carry them out every time the task has to be performed. It would be better if the system could carry out the overall task automatically. The idea of a Distributed System is that a scattered collection of resources and services appears to the user as a single, unified system capable of carrying out a particular task or type of task.

This illusion is maintained by the network software, that is, by a combination of communications software and operating software.

A network that provides access to a scattered collection that is both large and rich in resources and services can be converted to a number of Distributed Systems for different purposes. Different users of the same network can make use of different Distributed Systems at the same time. One user can switch from one Distributed System to another as circumstances may require.

To illustrate this, the Internet supports, among many other things, e-mail and the World Wide Web, both of which appear to their users as single, coherent systems. At the same time as one person is using the Internet as an e-mail system, another can be using the World Wide Web, and can perceive a quite different system. Similarly, a single user can switch from e-mail to the Web, and so have quite different views of the Internet at different times.

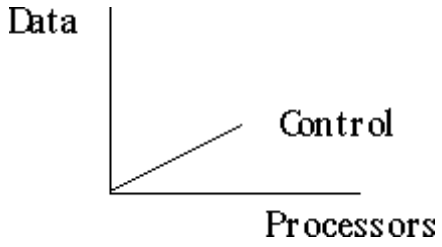
5. Examples of distributed system

This sections gives some of the contemporary examples of distributed systems.

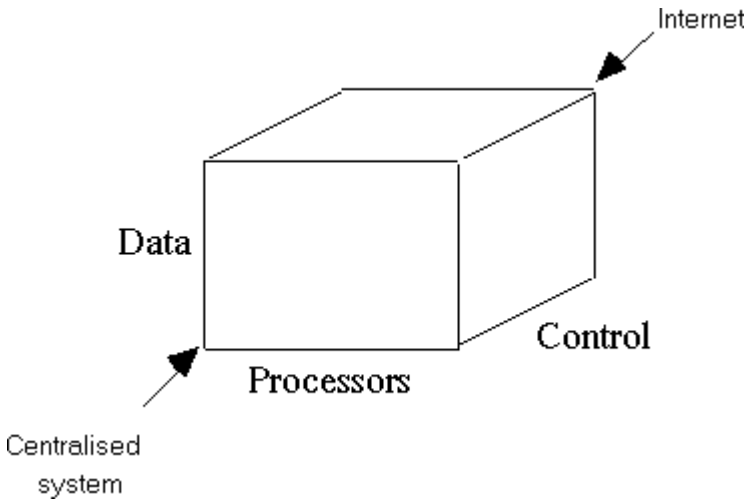
Domain	Examples
Search engines: Search engines are one of the most popular examples of distributed systems. The task of a web search engine is to index the entire contents of the World Wide Web. The engine crawls web content from different computers around the world.	Google, Bing, Yahoo and Baidu
Multiplayer online games: Online games allow multiple users to connect and interact through the Internet with a persistent virtual world.	Sony's EverQuest II and EVE Online from the Finnish company CCP Games
Financial and commerce: Financial and commerce systems allow users to trade online and provide access to information such as current share prices and trends, economic and political developments from multiple sources.	Amazon, eBay and Alibaba

6. The degree of distribution

It is useful to have a way of assessing the extent to which a network, and a collection of resources, is scattered, so as to be able to gauge the magnitude of the task involved in converting a scattered collection to a single system. One way to do this is provided by Ensloe's model in which a system can be located in three dimensions according to the extent of its distribution by considering the degree of distribution of its processors, its data and its control, as follows:



A system in which there is no distribution, that is, a completely centralised system such as a mainframe computer accessed by terminals with no significant computing or storage capabilities, will be located at the origin of the axes. The Internet provides us with an example of a system which is distributed world-wide in all three aspects. Since it is distributed to the maximum degree, it can be located on a cube such as that shown with sides whose lengths correspond to the maximum degree of distribution. It will be located on the corner of this cube diagonally opposite to the origin.



Any other systems will be located at some point on or in this cube according to the degrees of distribution associated with its processors, data and control.

So, with a means of assessing the extent to which a system is actually distributed, we have a way of assessing the extent of the task to be undertaken by the software that must turn this system into something that appears to its users to be a single monolithic system.

7. Inherent properties of a distributed system

Distributed systems have certain natural characteristics. They possess the collection of resources held by the computers of the network. The connectivity of the network makes it possible for anyone with access to the network to use and share these resources. The redundancy of the network itself and, sometimes, of the resources, makes the system tolerant of faults. Because different resources can be used at the same time, it is possible to carry out activities concurrently whereas, if they were done by a single computer, they would have to be performed successively. These characteristics can be distinguished as follows:

- **Resource sharing**: The resources available on a network can be shared by the users of the network because the connectivity provided by the network enable all its users to connect to any of the resources. A resource may be used by itself to satisfy a need, or it can be used as a component of a larger system which, when configured from this and other available resources, will also carry out some required task. A sever on a local-area network holding the software needed by the networks community of users is a prime example of a shareable resource.
- **Independent failures**: In a distributed system, shared resources are scattered on different computers. A crash of hardware or software in one computer does not crash the entire system. Each component of the system can fail independently, leaving the others still running.
- **Concurrency**: When a task is carried out by making use of several resources within the network, different parts of the task will be run on different computers of the network. If two parts are independent of each other, it is possible for them to run not only on different computers, but also at the same time. This means that the overall task can be completed more quickly on a distributed system than it could on a single processor, where the parts would have to be carried out sequentially, and so more slowly.

Consider, for example, the task of evaluating the expression:

(2+3)*(4+5)

A single computer would first add 2 and 3 and store the resulting 5. Then it would add 4 and 5 and store the resulting 9. Then it would multiply the two stored results to get 45. With a distributed system possessing resources capable of addition, the first addition could be sent to one of these resources and, at the same time, the second addition could be sent to another. When the results are returned, they can be multiplied together. In this way, if we ignore the time needed to communicate, the single computer takes longer to perform the operations additions and a multiplication, while the distributed system takes the shorter time needed for one addition and one multiplication.

8. Desirable properties of a distributed system

Besides its inherent properties, a distributed system needs other properties if it is to appear to its users as a single, coherent system. These properties must be provided by the operating software that controls the operation of the distributed system. These properties include:

8.1. Heterogeneity

The services accessed in a distributed system run over heterogeneous collection of computer hardware, networks and computer software. The internet uses protocols to enable heterogeneous computers to communicate. Although computers implement the Internet protocols, they do not necessarily all provide the same application programming interface to these protocols. Therefore, a distributed system must be designed to handle these differences.

8.2. Openness

The mode of operation of the system, and especially of its interfaces, should be openly described. This is needed to ensure standardisation, and to provide developers with the knowledge necessary to make computers and resource providing devices that can be attached to the network. Developers can then make equipment that is compatible with the system, and can be attached to the system and will work at once. This, in turn allows the system to develop, grow and evolve.

8.3. Security

Shared resources in a distributed system are scattered on different computers and accessed using a computer network. Therefore security of resources is of considerable importance. A distributed system must provide three types of security:

- Confidentiality: protection against disclosure to unauthorized individuals.
- Integrity: protection against alteration or corruption.
- Availability: protection against interference with the means to access the resources.

8.4. Scalability

The system must operate effectively regardless of its scale. If a small-scale system evolves to a larger scale, it must maintain the same behaviour so that its growth is not apparent to its users. So, for example, a system should give more or less the same level of performance as it evolves in size and complexity. The Internet is a good example of a system that has maintained its level of performance and overall behaviour as it has grown in size and complexity.

8.5. Transparency

The whole point of a distributed system is that its users should ‘see through it’ to be able to do what they want without being concerned with how it is done. Matters such as taking advantage of concurrency and dealing with the scale of the system must be attended to automatically and in a way that is not apparent to the users of the system. The system should not be transparent to location, so that users do not need to know where a resource is located; to failure, so that they do not notice when the system is working around a fault; operation, so that they do not have to know (at a technical level) how the system works or even if the way it works is altered, and so on.

9. Techniques that ensure the desirable properties

There are some well-known techniques that can be used to ensure that a distributed system possesses the desirable properties described in the previous section. They include:

1. **Naming**: when resources are named, each can be accessed merely by giving its name. This removes the need to know where the resource is located and allows all the sharable

resources to be treated equally as members of a common pool. As with addressing systems, naming systems may or may not be scalable.

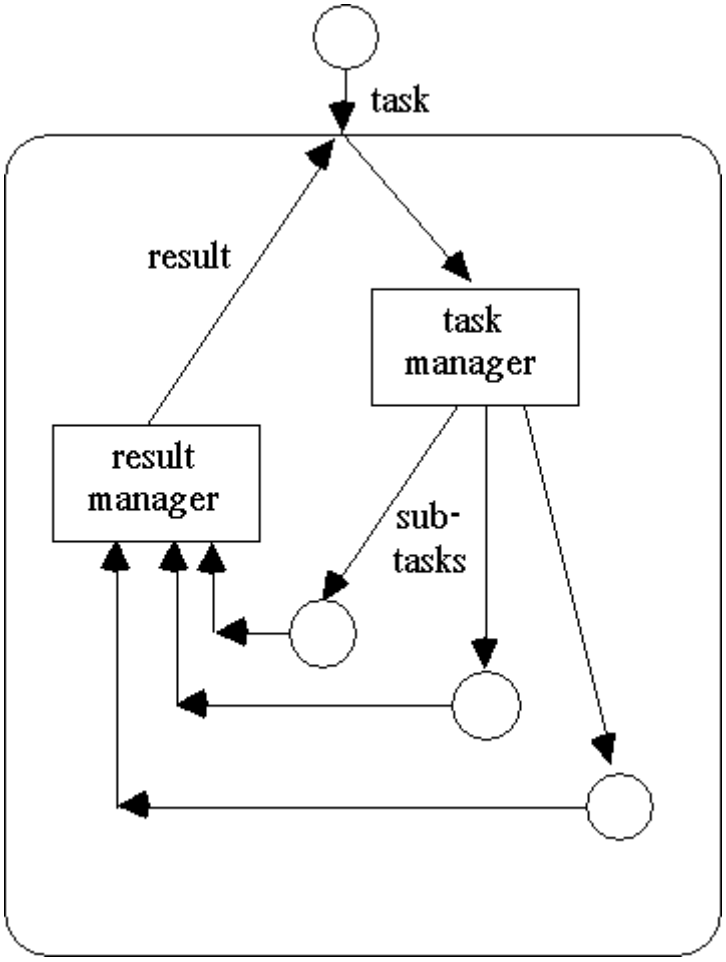
2. **Software structure**: One way to ensure an open system with a clear description is to ensure that the software used in the creation of a distributed system has a clear and clean structure. The standard, publicly available description of the system can then be clearly and comprehensibly described.
   As we have seen with the Internet software, layering is one way of structuring software. Layering not only sub-divides the software according to function but also has the added benefit that the positions and descriptions of the interfaces can be clear. This provides developers with just the information they need to develop compatible products and, what is more, allows for its provision in a clear form.
3. **Communication and work allocation**: Communication across the network must be sufficiently rapid and reliable as to be unnoticeable. The requisite patterns of communication, such as client-server exchange, must also be readily available. The system software must be able to allocate tasks and sub-tasks to network resources in a fair and transparent fashion. This may require replication of resources and services which, in turn, will require the maintenance of consistency between replicated items.

10. The pay-off

It is not easy to create a Distributed System, but the rewards for doing so can be considerable, especially if it can be based on the Internet. Users can be provided with powerful and flexible systems that can be used in a straightforward and uniform manner. Service providers have a framework for the provision of services, and a platform for their wide-spread access.

The essential reason for this is that a Distributed System can support, in a transparent way, a collection of resources. These resources can be hardware, and may include any kind of computer or peripheral device. They can be software, including programs for processing information, and information itself. Since any task can be accomplished as long as it has access to the necessary resources, a Distributed System is in a position to carry out the tasks of its users by providing the resources they need from the pool of resources that it supports.

When the resources of a Distributed System include a resource manager capable both of determining the resources needed for a particular task and of making them available, a user of the system need only send a task to the resource manager in order to have it automatically carried out. If the system possesses the resources needed to carry out communication tasks, it becomes a communication system; if it possesses the resources needed to carry out the tasks of a digital library, it becomes a digital library; if it possesses the resources needed to carry out tasks of the market place, it becomes a market place. The following diagram illustrates this point by illustrating schematically how a task presented to the system may be carried out automatically by mustering and organising the requisite resources.



In this way, the concept of a Distributed System explains why it is possible to have many different views of the network on which it is based. The concept also provides a framework within which these different points of view may be examined in a coherent fashion.

11. Activities

You can find a discussion of these activities at the end of the chapter.

Activity 1 - LAN

Assess the degree of distribution of a typical local-area network by positioning it in the appropriate place on the Ensloe cube. Assume that the LAN can support a maximum of 40 computers, that it is contained within a single building and that it has a single sever.

Activity 2 - MAN

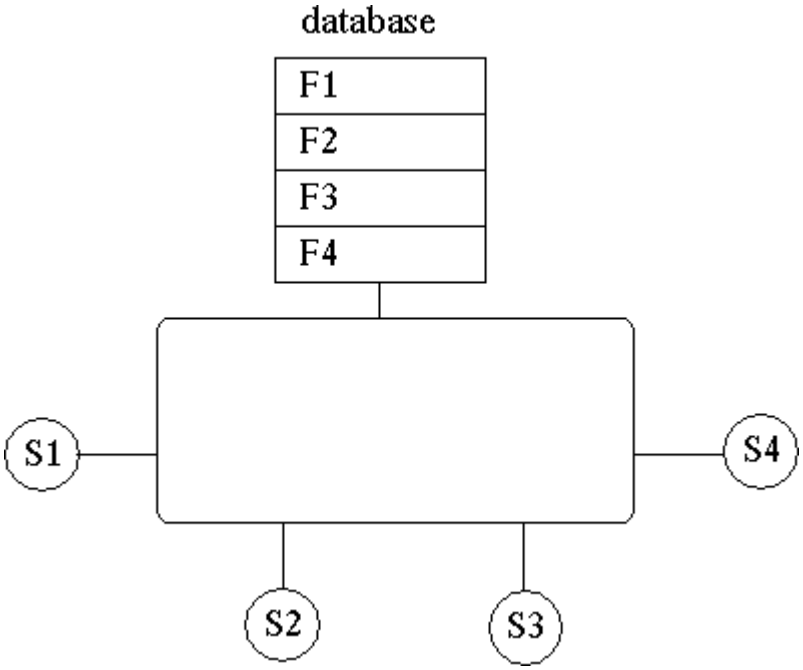
Assess the degree of distribution of a typical metropolitan-area network by positioning it in the appropriate place on the Ensloe cube. Assume that the MAN has a ring topology and that its coverage is city-wide. Its purpose is to link LANs such as those in Activity 1 that are located throughout the city.

Activity 3 - Concurrency

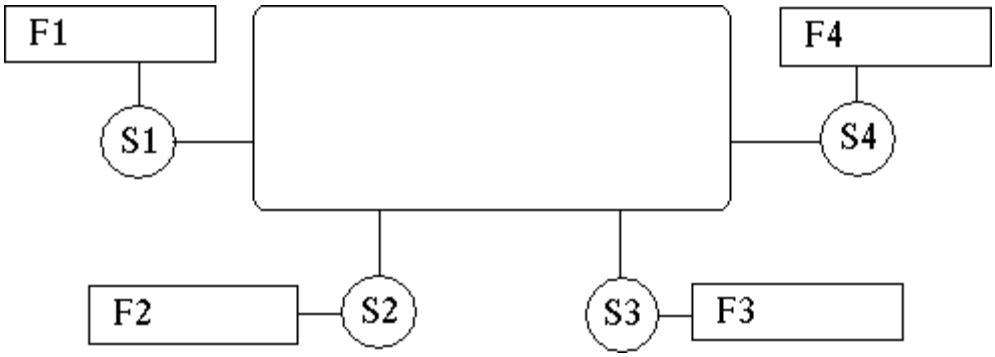
Explore and explain ways in which the kind of search carried out by a search engine when searching

Activity 4 - Databases

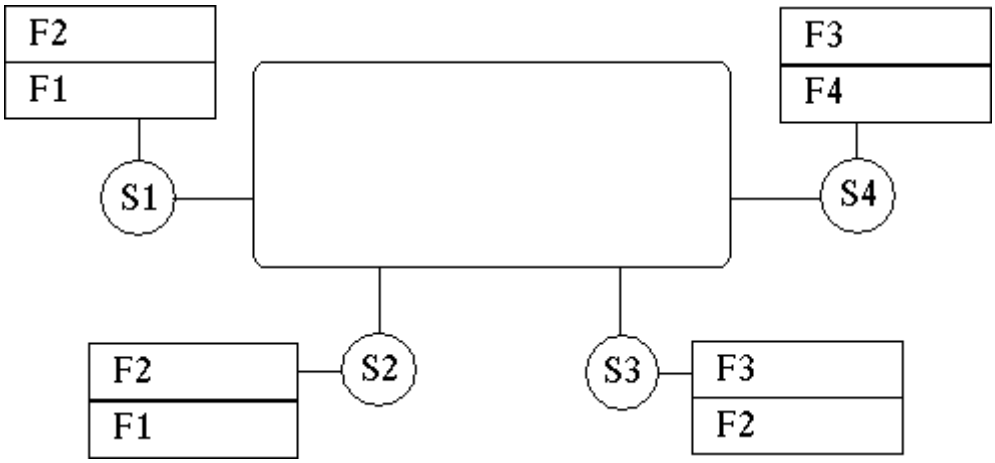
An organisation operating on several sites, S1 to S4, could make use of a central database to hold its staff records. In that case, the situation could be represented as follows:



The organisation might find it preferable to make use of a distributed database. Suppose the database can be divided into four fragments, F1 to F4, so that F1 holds the details of the staff based at S1, and so on. This situation can be represented by:



It may be however, that, although staff are based at one site, they work not only at this site but also at another one. In that case, it might be of value to replicate the fragments, as follows:



Compare the two distributed database scenarios with the centralised scenario so as to:

1. Bring out the organisational reasons for favouring distribution,
2. Bring out the technical reasons favouring distribution,
3. Bring out the technical problems involved in ensuring that the distributed database is a Distributed System. That is to say, that although it is actually distributed in one of the ways shown, it appears to its users to be just like the centralised system shown.

Activity 5

Using the diagram given above as a basis for explanation, identify the role of the task manager, the subtasks and the role of the result manager when the available resources are mustered to send an e-mail.

12. Review Questions

You can find an answer/comment for these review questions at the end of the chapter.

Review Question 1

How would you define a Distributed System?

Review Question 2

How is the illusion of that a Distributed System is a single monolithic system created despite the actuality of its being geographically scattered?

Review Question 3

What is the essential purpose of a Distributed System?

Review Question 4

Can you think of another type of network with a degree of distribution lying somewhere between the extremes provided by the degrees of distribution of a centralised system and the Internet?

Review Question 5

List the resources needed by e-mail so that it can operate in the way to which we are accustomed.

Review Question 6

Give one example of a shareable software resource and one of a shareable hardware resource.

Review Question 7

Why, in general terms, is it hard to ensure that a system maintains its performance and appearance as it increases in scale?

Review Question 8

What is the consequence of a system not being transparent?

Review Question 9

How is openness assured? What are its benefits?

Review Question 10

Which of the desirable properties listed above follow from the use of naming?

Review Question 11

Which of the desirable properties listed above is facilitated by the use of clear software structuring?

Review Question 12

Which of the desirable properties listed above is facilitated by the use of work allocation?

13. Discussion Topics

1. Are the techniques described above sufficient to ensure that a Distributed System has all the properties that it is desirable for it to have? If not, can you find further techniques in Coulouris to cover the gaps?
2. In designing a distributed database, issues that need to be addressed include:
    - How to fragment the database,
    - how to allocate fragments to physical locations, and
    - whether and how to replicate and allocate fragments.

    The decisions made concerning these issues affect:
    - information retrieval,
    - database integrity, and
    - the balance between communication and computation.

    Expand on this in terms of the technical issues that must be hidden from the users. Look at it again, this time in terms of a trade-off between computation and communication (and perhaps also in terms of the trade off between the cost of computation and the cost of communication).
3. How will a discussion parallel to that for designing a distributed database proceed when the subject is the design of groupware?

14. Answers and Comments

Activity 1

The processors are all in a single building, so that although they are scattered within the building, by comparison with the Internet, they are not widely scattered. This gives a position on the cube at a small distance along the processor axis. Control is distributed among the processors, so that control is distributed to much the same degree as the processors themselves. If we assume that the data is all held on the server, then it is not distributed at all. This gives a position at the origin of the data axis. Of course, on some LANs, some of the data may be held at the client machines, and so the position for a LAN like this will be slightly different.

Activity 2

Here, the processors are distributed on a city-wide scale, as is the control. The data is scattered across the various servers on the LANs, as well as any storage resources that the MAN itself may provide. Thus, the position for the MAN is further from the origin than that for the LAN, and is also more deeply inside the cube.

Activity 3

Given the size of the Web, to search it serially for a particular item is almost inconceivable. Such an approach is not viable given the size of the search space, and the time it would take to search it.

However, if the search space can be divided into parts, each of which can be searched at the same time, the problem of searching the entire space may become possible.

Activity 4

1. An organisation could find it more convenient, efficient or appropriate to keep the records of the staff at the site where they are based, so favouring fragmentation. If similar benefits follow from keeping the records of staff working on a site at that site, then fragmentation and replication would be beneficial.
2. A centralised system is neither fault-tolerant nor concurrent. In fact, a centralised system is highly vulnerable to failures whereas either form of distribution will support concurrent activity at the local sites.
3. The purely fragmented solution will work well as long as database queries refer only to records based on site. As soon as a query refers to anything larger than the locally based fragment, including the entire database, the Distributed System software must automatically and transparently deal with that situation.

The solution involving fragmentation and replication must deal with this problem and also with issues involving the integrity of the database.

Activity 5

The task manager usually resides at the e-mail submission server, and co-ordinates the sub-tasks of converting the e-mail address to an IP address, and submitting appropriately addressed packets to the Internet for delivery to their destination. The result manager, residing at the same place, will inform the user when the e-mail has been successfully sent.

Review Question 1

A Distributed System can be defined as a system that is actually geographically distributed but which appears to its users as a single, coherent system.

Review Question 2

By adding software of the right kind to the hardware.

Review Question 3

To make available in a transparent way the resources it possesses.

Review Question 4

Networks with intermediate degrees of distribution include local-area networks, metropolitan-area networks, university networks and national packet-switched networks.

Review Question 5

It needs, at the least, a resource to convert e-mail addresses to IP addresses, and a resource to deliver the e-mail to its destination.

Review Question 6

Any networked archive or database is a software resource that can be shared. A network printer and a network storage device are examples of hardware resources that can be shared.

Review Question 7

As a system grows, it increases in complexity. With more component parts, there are more possibilities for interaction, and more chances for things to go wrong. As the number of users increases, the demand on popular resources increases and, unless steps are taken, the quality of service from those resources will decline. To maintain performance, the system itself must be changed, for example, by replicating much-used resources, but this must be done in a way that is not apparent to the users of the system.

Review Question 8

If a system is not transparent, so that its users cannot 'see through it', then those users will have to be aware of how the system works. Rather than just using it, they will have to learn something of its internal workings in order to make use of it.

Review Question 9

Openness is assured by publishing an account of how the system works. If the system is designed in a clear and structured manner, then that account can also be clear and structured. The benefits of this openness are that anyone who wants, or needs, to understand the operation of the system can do so. This includes developers of equipment for the system, developers of software, and those interested in improving it.

Review Question 10

Naming makes possible the transparent access to resources.

Review Question 11

Openness.

Review Question 12

Automatic work allocation ensures that the appropriate resources can be allocated transparently to the task at hand.