What is a System?

The term “system” is derived from the Greek word “synistanai”, which means “to bring together or combine”. It is beyond the scope of this article to embark on a detailed discussion of system theory. However, it is useful to have a working definition of the word ***system***.

If we look at some examples of systems:

- Solar system
- Digestive system
- Public transport system
- Central heating system
- Computer system

We may arrive at a tentative definition – *a system is a set of objects or elements that are viewed as a whole*. On its own, this is inadequate for our purposes because we are concerned only with systems that are man-made, and therefore under human control, and that have a purpose – otherwise the system cannot be designed by a system developer. This rules out the solar system (no known purpose) and the digestive system (not under human control). An improved definition therefore would be: *a system is a set of objects or elements that are viewed as a whole and designed to achieve a purpose.*

Relationships

Another essential feature in our view of systems is that the elements of a system have a relationship to one another; they work together in some way. A heap of stones, for example, although it may be man-made and have the purpose of marking the top of a hill, does not qualify as a system because the elements do not have a significant relationship to one another. If you take one stone from the pile, it does not matter much to the others. In a system removing one element would matter. If you remove the train service from the public transport system, it puts pressure on the other services – it affects them. If you remove the boiler from the central heating system, the system will not work. For our purposes, therefore, the definition of a system we need is: *a system is an interrelated set of objects or elements that are viewed as a whole and designed to achieve a purpose.*

Boundaries & Environments

We must add to this definition that a system has a boundary. The system in question lies inside the boundary; outside the ***system boundary*** is the ***environment*** with which the system interacts. Sometimes the boundary of a system is clear and obvious. If we view a person as a system the boundary is clear – one person is clearly separate from another and from the environment. In computer systems, however, it is usually hard to define the boundary; it is dictated by which elements we choose to think of as being within the system, and which as being part of the environment. The normal rule is that inside the system are things the system is designed to control;

outside the system boundary are things the system interacts with, but is not designed to control. The boundary may be set because there are things over which we cannot have control – for example, in a central heating system the weather must be considered to be outside the boundary as we cannot control it. The boundary may also be set because we choose not to include certain elements. This choice may be dictated by the following factors.

Money constraints.
We may find that it will cost too much to computerise more than a limited set of system functions.

Time constraints.
The more functions we computerise the longer it will take.

Cost effectiveness.
Sometimes only limited benefits are gained from expensive computerisation.

The environment is defined as being the surrounding conditions, outside the boundary, that affect the system and may be affected by it but not controlled by it. We might define the weather conditions as being part of the environment of a central heating system.

Summary

To summarise: a *system is an interrelated set of objects or elements that are viewed as a whole and designed for a purpose; it has a boundary within which it lies and outside of which is the environment.* It is important to define the purpose or objectives of the system; different users will want different things from the system. From the outset, the system developer must be clear about the purpose of the system.

CAROL BRITTON & JILL DOAKE (2006) SOFTWARE SYSTEM DEVELOPMENT
4TH EDITION, MCGRAW-HILL