CHAPTER 4
STUDY ON INFORMATION MODELLING

4.1 Introduction
A model is some description of a system and it is used to represent the complex structure of a system or a process. This chapter explains principles of modelling and its advantages. To develop a model, many modelling methodologies are available. Since modelling of feedback control is a complex process, it will not fall into a single methodology. So information modelling, process modelling and hybrid modelling methodologies which are necessary are discussed here.

4.2 Principles of Modelling
A model is a communication of structure and properties. It usually presents itself as representing something, either in the 'real' world, or in the intentions of people (managers or engineers, planners or designers). Sometimes a model merely represents itself. A model always has a scope, purpose and perspective. Sometimes these are stated explicitly. Often these are implicit or assumed [72].

Any model is an abstraction; i.e., it ignores certain state variables of the system being modelled and represents other state variables. The criterion for deciding which state variables to represent is the perspective. Thus if a person wishes to use the model rather than the reality in order to ask questions concerning some property P because with respect to that property (or for that purpose) the model is as good as the reality, then P is the perspective of the model. This may be redefined and renegotiated during the project. Identification and confirmation of stakeholders is one important task [70].
Formal models are expressed in a strict modelling language or notation. The vocabulary and grammar of this modelling language may themselves be expressed as a model, which software engineers usually call a metamodel. Such models are used for building support tools, such as diagramming tools and repositories. For many purposes, informal models are as good as formal models - indeed, sometimes better [70].

4.3 Modelling Purpose

The modelling alters cultural attitudes of process management in an organization. The existence of the model itself changes the situation, by altering attitudes or surfacing anxieties/assumptions. The modelling adds the stock of generalized and reusable knowledge. This motivation has three possible perspectives: an organizational perspective, an industry perspective and an academic perspective. The modelling helps to take informed action. The models are used to support design decisions, investment decisions, procurement decisions and planning decisions [72].

Furthermore, the modelling process may have some beneficial features [71]:

1. Where different stakeholders have different views of a situation, these differences can be represented and discussed.

2. Including something in a public model legitimizes it as a topic of discussion - the model itself 'contains' the anxiety of discussing taboo subjects.
3. Modelling is a way of exploring what the requirements might be, and whether such generated requirements fall within the scope of the system (as currently defined) or whether the scope has to be changed in order to encompass them.

4. The model supports or stimulates organizational learning, and also allows the participants/observers to develop knowledge that may be applicable elsewhere.

4.4 Modelling Methodologies

4.4.1 Information Modelling

The main purpose of the information model is to define application entities and data at the conceptual level. The information model must at least, include components by which we can describe the contents of the different information sets and flows of the Business Processes Model [58]. Consequently, the information model includes components such as entities, binary relationships and information attributes. Also the ISA and PartOF relationships are included in the information model in order to permit generalisation as well as complex component modelling. The information model allows also the possibility of defining different "Information Model Component Groups". A group of this type is simply a view of a part of a information model, and includes a subset of the information model's entities, relationships and attributes. A group can be a member of another group, etc. Groups may overlap each other in terms of their components. Defining a language for the information mode is based on an easy to understand conceptual modelling language. However, the information model may also be defined using the more powerful ERT modelling language, e.g. where temporal properties of entities as well as of relationships can be described. The choice
of modelling language does not affect the modelling process itself. It is generally possible to begin with the notation that described in the manual and then, later on, switch to more advanced information modelling language. This might be particularly important when information model is used as a requirements source for a database design [58].

Information modelling is an outgrowth of data modelling. Data modelling is concerned with specifying the appearance and structure within the computer system of data which represents particular types of information. Information modelling has a goal of describing information so that representative data could be computer processed. One distinction between data and information modelling is that one is explicitly targeted for computer processing of the data while the other has a potential for such processing. The other distinction is in treatment of interpretation rules. In information model these must be made explicit and formally documented. In a data model, the rules are typically implicit; even if they are made explicit, they are informally documented.

Data analysis/information modelling has been practised in the IT industry for many years, for a number of related purposes: database design, information system design, data warehousing, and system integration and migration. Despite the recent popularity of object technology and object modelling, and despite the growth of component-based development (CBD) and open distributed processing (ODP), data modelling remains an important technique for software developers - regardless whether these models are expressed as entity-relationship diagrams, in the data modelling extension of UML, or in some dialect of XML. Although information modelling remains a key
component of traditional top-down structured methods and database design, it finds
good practical use in many other types of project. For example, to map data between
interfaces and between systems, logical and physical data models representing the
interfaces are needed. All of the following need some element of data modelling: data
warehousing, data mining, data migration, enterprise application integration, system
merger, legacy mining, B2B, BizTalk, web services. To use a web service or a
software component, one have to adopt its vocabulary, or map its vocabulary to ours
and these are both essentially data modelling activities [69].

4.4.2 Process Modelling

A process is a set of inter-related work activities. These activities require specific
inputs from one or several suppliers, perform some specific value adding tasks on
those inputs and deliver specific outputs to some customers. The term “suppliers” and
“customers” refer to data sources and sinks (internal and external to the system
analysed) or it can also be other individual processes. A process may be fully
contained within one organization or span several organizations. There are different
types of process models – Descriptive modelling and Active modelling [21].

Descriptive process models are just that models which are some way describe
organizations in terms of processes. Such models may be formed in a variety of ways,
using a plethora technique. Generally such techniques will be supported by software
tools which enable the modeler to create the models. A number of specific techniques
have been developed to support the production of models. Active modelling gives
more information about the idea that process models can be used to provide computer
based support for systems [73].
Process models can be used to positively influence an organisation in several ways: Process models can be used to cultivate the shared awareness of the required activity and priorities, and reduce the loss and distortion of messages that happens as instructions are passed through large organizations. If the model is clear it can bring the same message about what is needed to everyone from the chief executive to the technicians in the field and they can each see what role they must play in the model. If there is no clear idea about what is going to be improved then there will be no productivity or performance improvements. A process systematically modeled will allow identification of individual activities at a level where their value can be assessed. Then it is possible to decide if all activities are necessary and/or if some of the activities can be done in a better way [21].

The same model allows stakeholders to determine what metrics will truly represent the performance of the process and to locate where bottlenecks are occurring and take appropriate action. Additionally the model can help organizations to make more accurate schedule and resource plans as they provide clearer views of what needs to be done at any point in a process. A well modeled process allows less experienced staff to be led through an operational procedure instruction (much in the same way that on-line help in Windows 95 lets everyone sort out their PC). This method could be applied to any process where the process model itself can be the basis of the step-by-step guide. A help-line operator interacting with a customer could simply call up a help box about, say, subscriptions to new services which would indicate the required steps to go through, what information must be recorded at each step and who the information must be sent to in order to give customers what they want [21].
The obvious consequence to having a given process modeled and to having operational procedures on-line would be to automate the parts of the process that require little intellectual input. Simple tasks like electronic distribution of components of a service provision request to, for example, field technicians could be done efficiently by modern data communications. Thereby the printing of multiple copies of a work order containing all the details of the service request, posting copies to everyone and expecting copies to be posted back. The point here is that the model allows a clear view of the parts of a process that would lend themselves to automation [21].

According to the object-oriented view, everything can be understood as an object. In particular, processes can be viewed as objects. This is known as reification. This leads us to understanding the world simply as a system of interacting objects [21].

In any case, the object-oriented view fails to support the way that business people want to think about business. A business is not just a group of connected things. It is not even just a group of connected people. It is business processes and relationships [21].

A process can be a model from several different perspectives. A process can be described from several different perspectives, according to the purpose of the model. For example, a video-on-demand process might involves many different companies, controlling many different information flows through many different technical mechanisms. The processes flows would be defined either in terms of information flows or information states [72].
4.4.3 Hybrid Modelling

The combination of two or more modelling methods is called hybrid modelling. Hybrid systems have been the topic of intense research activity in recent years, primarily because of their importance in applications. These include hardware and embedded software verification, mobile communication networks and large scale, multi-agent systems, such as air traffic management, automated highways, power networks, and Uninhabited Aerial Vehicles.

The advantages of the hybrid modelling technique [60] are

It allows to choose the most appropriate or natural modelling language matching the characteristics of each domain. Because each language is only used to express the elements of one of the domains, the expressive power needed is relatively low. This results in relative or analytical complexity, while the sacrifice in the accuracy of performance prediction is limited.

It makes it easier to relate analysis results to performance measures of the original processes. Performance measures from the behavior domain e.g. process completion times and customer response time and measures from the entity domain, e.g. business resource utilizations, follow more easily from the analysis of a hybrid model than from the analysis of monolithic model. A related advantage is that such a hybrid techniques facilitates the use of performance analysis in combination with scheduling (e.g. to provide the performance measure used as optimization target in scheduling), because similar concepts are used (tasks which are mapped onto resources).
It enables the reuse of aspect models. For example, when using performance predictions to compare alternative implementations of business process with given business resources, only the behavior model needs to be changed; the same entity model can be used for all alternatives. Conversely, when comparing different organisation structures to perform the same business process, the behavior model can be reused.

The software process is composed of various activities (e.g., the development of the specification document, the change of the specification document following requirements changes, the correction of the design document on the basis of defects discovered during the testing, etc.). Some activities are sequential, others may be carried on concurrently. Activities collaborate to develop the final product through the exchange of artifacts. Activities need resources (e.g. personnel, computer time, etc.) to carry on the required tasks and may collide on the use of the same resources. The software process does therefore show both discrete system aspects (start/end of an activity, reception/release of an artifact by an activity) and continuous system ones (resources consumption by an activity, percentage of developed product, percentage of discovered defects).

The hybrid two-level approach introduces two abstraction levels, a higher abstraction and a lower abstraction level, to represent the software process by a combination of three methods: the analytical, the continuous and the discrete-event method. At the higher abstraction level, the discrete-event modelling method is used. The process is modelled by a discrete-event queuing network, which models the component activities, their interactions, and the exchanged artifacts. The queuing model is a
direct replica of the software process, with service stations used to represent activities, and circulating customers used to represent artifacts that move from one activity to another. Each activity can on its turn be described by a set of service stations to represent the component sub-activities. Some activities/sub-activities may consume resources (e.g. personnel, time), whereas others may just perform co-ordination tasks, simulating managerial policies (e.g. whether or not to release a process artifact to the next development stage depending on its quality level). The lower abstraction level gives the implementation details of the service stations (i.e. activities) introduced at the higher level. Here the analytical and continuous methods are used. In particular each activity is modelled by either an analytical average-type function, or by a continuous type time-varying function, or by a combination thereof. Such functions are used to express the amount of resources, or time, or effort that various service stations use to simulate the corresponding activities or sub-activities [68].

4.5 Discussions

It would be impractical to require a feedback control information model to be fully defined before implementation. Hence dynamic modification or extension is an essential feature of the information model for the feedback control. In order to implement an information model for feedback control, it would be necessary 1. to provide an ability to dynamically evolve the schema at runtime and 2. to have a database management system that allows on-line schema changes while applications are running.

It is clear from the literature that there is not any information model for the feedback control process. Meta-modelling method used in the software engineering field is
good in integrating data models, and a model with a meta knowledge has been identified for designing integrated information models that will aid in building information systems that can be dynamic, co-operative and distributed. Since the information model for feedback control involves various information such as structural and instances, combination of models is necessary to solve the problem.