The Design of Service Support Systems With Respect to Reliability

A thesis submitted to The University of Manchester for the degree of

Doctorate of Philosophy

in the Faculty of Engineering and Physical Science

2007

Gong Wang

Extreme Loading and Design Group

School of Mechanical, Aerospace and Civil Engineering

List of Contents

5 'Bottom Up' Service Design

Appendix B Simulation Models of Typical Activities

List of Tables

List of Figures

List of Equations

List of Abbreviations

AA	Automobile Association
CAD	Computer-Aided Design
CIT	Critical Incident Technique
CHDS	Centralized Heterogeneous Distributed System
FPs	Functional Products
FMEA	Failure Mode and Effect Analysis
FTA	Fault Tree Analysis
GSSS	Generic Service Model
$IDEF_0$	Integration Definition for Function Modeling
ISO	International Organization for Standardization
KPI	Key Performance Indicator
NSD	New Service Development
PC	Power Contract
PERT	Project Evaluation and Review Technique
PSS	Product Service Systems
RBD	Reliability Block Diagram
QFD	Quality Function Deployment
RPN	Risk Priority Numbers
SADT	Structured Analysis and Design Technique
SDI	System Design Interface
SSS	Service Support System
TLS	Total Landing Gear Support
VAC	Volvo Aero Corporation

Nomenclature

AOG	Aircraft on Ground
A_{Opt01}	Output 1 of activity A
A_{Opt02}	Output 2 of activity A
B_{Opt01}	Output 2 of activity B
C_a	Resource prescribed for an activity (person)
$C_{a,actual}$	Actual resource prescribed for an activity (person)
C_o	Cost of outages
C_{Opt01}	Output 1 of activity C
C_s	Service cost (person)
N_r	Number of repeat times
N_{rmd}	Random Number
P_a	Resources consumed of an activity (person)
P_S	Total resource consumed by the SSS (person)
R_a	Activity reliability
R_s	System reliability
S_{Ipt}	System Input
S_{Opt}	System Output
S_{Opt01}	Output 1 of system
S_{Opt02}	Output 2 of system
S_{Opt03}	Output 3 of system
t_a	Activity time (hour)
T_a	Accumulated Time taken by an Activity (hour)
T_d	Standard Deviation of Activity Time (hour)
$T_{m,actual}$	Actual Mean of Activity Time (hour)
T_{mean}	Mean Activity time (hour)

T_S	Total time taken by the SSS to generate the system output (hour)

The Design of Service Support Systems With Respect to Reliability

Doctoral Candidate: Gong Wang
Doctoral Advisor: Professor Graham Thompson
School of Mechanical Aerospace and Civil Engineering
University of Manchester, UK

Abstract

Service reliability is emerging as a new focus for research in the field of reliability. New products are being designed that are a combination of hardware plus a service support system. The service support system not only keeps the hardware functional but also provides opportunities to improve the performance of hardware. In such new products, known variously as product-service systems, functional products and total solutions, service reliability has a significant influence on the success of the complete product. A design support tool that can be used to evaluate service system reliability during the design process would be useful and this research aims to fulfil this need.

A review of literature in the field of reliability as applicable to service design and a review of the state of art of service design are given. The failure characteristics of services are discussed and a method of service modelling is presented. Two approaches to service design are proposed which enable service reliability to be considered. The approaches encompass a computer-based simulation that enables designers to work with clients to create a system to fulfil client needs. The method has the capability to evaluate the service with respect to reliability and to identify particular reliability-critical activities. Modelling and simulation determine the resources used and the functional reliability of the service system, the probability that the system will meet the contractual service time requirement. It also can be used to identify internal failures in a system. It is therefore possible to improve the system performance by allocating resource to key system activities. The thesis demonstrates that computer-based simulation is an effective method to evaluate service reliability by carrying out a case study concerning the design and evaluation of a comprehensive service support system for aircraft engines.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

I would like to express my appreciation to Prof. G. Thompson for his advice during my research for the past four years. As my supervisor, he has constantly motivated me to focus on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with investigation in depth.

I am grateful to Dr. T. Alonso-Rasgado she has helped me greatly and been a source of knowledge.

I would like to express my sincere thanks to my peers Dr. Jian-Ping Li and Ryan Chan for generously sharing their time and knowledge.

Last, but not least, I would like to dedicate this thesis to my family, my wife, for their love and support.

Gong Wang
Nov 29, 2007, Manchester, United Kingdom

Chapter One

Research Background and Objectives

This chapter provides a high-level overview of the application of a computer-aided service design process with respect to reliability. It briefly presents the research background from which the idea originated. The research objectives are outlined, and the chapter ends with an introduction on the organization of the rest of this thesis.

1.1 Research Background and Inspiration

Extensive evidence indicates that the service industry is booming and playing a more and more significant role in the economies of developed countries. Tortorella (2005) stated that service industries are becoming the major part of the economy in many developed countries. A similar belief is expressed by Hollins (2006) who stated that in industrialized countries worldwide an increased contribution to the GDP and to the level of employment is derived from non-manufacturers, or more especially, the service sector. Moreover, the importance of 'services' to the economy of a country will almost certainly continue to grow in the foreseeable future. Contributory factors include general economic prosperity and smaller family sizes as higher disposable incomes have led to an increase in financial services, entertainment, eating out, travel, personal healthcare and fitness.

In addition to the prosperity of traditional sectors such as hotels, restaurants and banking, service is also starting to offer profitable innovation in manufacturing industries, for example the Product Service Systems (PSS) which integrate hardware and supporting services as total function offers; see figure 1.1. Other names used in industry for such products are Functional Products (FPs) and 'Total Care' products. The integrated supporting services are often referred to as service systems; in this thesis the term Service Support System (SSS) will be used. FPs have become a new hotspot in manufacturing industry. Suppliers of such products view them as a new way to maintain their business profitability and their products' competitiveness. Examples include the sale of 'power by the hour', the Total Care® engine provided by Rolls-Royce to aircraft operators rather than the sale of the aircraft engine hardware alone (see figure 1.2), and the supply and support of key process machinery that performs a defined function as part of a production process (Alonso-Rasgado *et al.*, 2004).

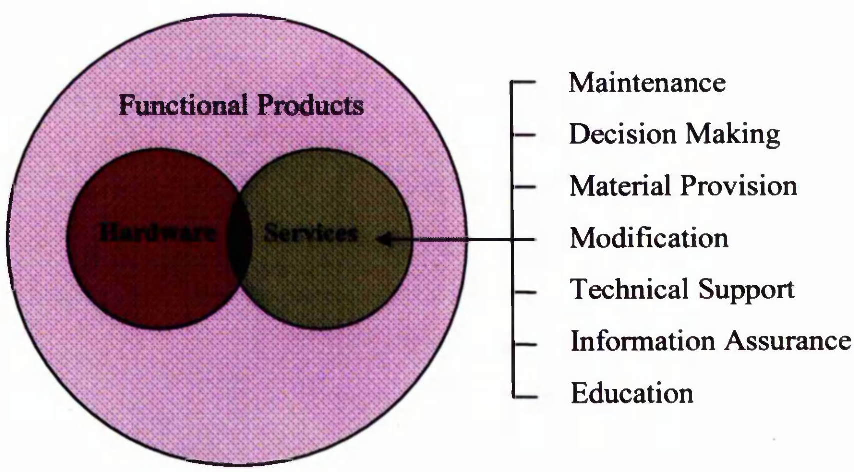


Figure 1.1: Functional Product (Adapted from Persson, 2004)

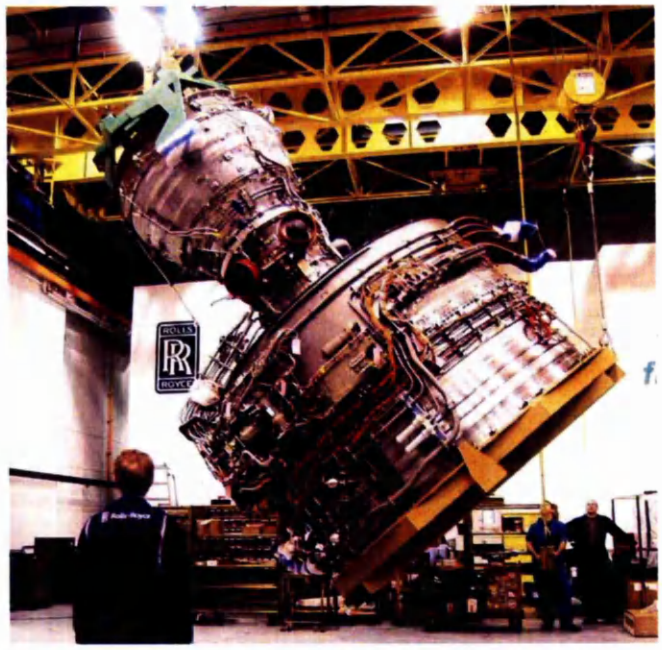


Figure 1.2: Total Care® engine provided by Rolls-Royce (source: www.rolls-royce.com)

The success of this innovation from the business perspective has been demonstrated by Rolls-Royce's rapid and substantial gains in market share over recent years. Moreover, it has been pointed out that profits from civil aircraft engines come not so much from sales of the units as from the supply of spare parts and services over decades of use. As *Travel & Tourism* (2003) reported, Rolls-Royce's engine deliveries have grown to a total of 54,000 gas turbines in service worldwide and annual sales of 'power by the hour' totalling around £6 billion, over 40 percent of which currently comes from aftermarket services.

Nevertheless, there exists a common challenge facing these FP suppliers. They still have to deliver the contracted level of availability of the core hardware, and therefore, need to look for ways to make their integrated offers more reliable. This situation was discussed by Bitran and Pedrosa (1998) who comment that in a world of increasing competition, manufacturing companies are being required not only to design better products but also to design appropriate support services. On the services side, companies are required to look for ways to make their operations more reliable, consistent, and replicable at the global level. As Hill

(2002) stated, although more companies claim that 'we sell solutions' including services intended to add value to the customer's use of tangible products and lower the customer's total life cycle cost, not all can survive this intense competition. Success is more likely to be achieved if a systematic, product-like approach to service development is employed.

Therefore, to achieve successful provision of FPs, the suppliers must consider the reliability of all parts of FPs, both the hardware and the supporting services. Hardware reliability has received much attention in the literature, but service reliability less so. Meanwhile, some other reliability-critical services like real-time telephone conferencing and power distribution are also required to be highly reliable. Some companies have therefore adopted the 'error-free' strategy to maintain their competitiveness, which to some extent gives rise to research into service reliability.

However, the reliability of products and systems will not increase automatically through quantified assessments. Davison (1994) thinks that reliability analysis and prediction, of itself, will not solve the problems of unreliability and poor availability, but will only provide information to form a basis for either further engineering investigation or rational decisions on whether to modify or replace components, re-design the plant system, or increase the levels of certain items of spare (redundant) plant. This viewpoint is supported by Thompson (1999) who considers it effective to improve reliability through design. Thus, improving reliability through design has become a critical concern of FP providers.

Nevertheless, for some time after the fuzzy standard of 'service quality' was established, there was no appropriate methodology for designing reliable services. Only recently has the significance of service design been recognized, receiving increasing attention. As Hollins (2006) claimed, only recently have managers in organizations involved in this service sector realized that a conscious effort in applying 'design' techniques to service can result in greater customer satisfaction, greater control over their offering and greater profits.

Unfortunately, there are few resources available that can assist these managers in the application of design to their service products.

Alongside the concern for reliability in designing services have come other design difficulties, including how to tailor highly customized services, that is, how to address customers' particular requirements more effectively, and how to achieve the optimal level of system performance with limited resources. This thesis will propose an innovative design approach which enables service providers to address these issues effectively.

1.2 Objectives

This research aims to be a contribution to creating an innovative service design approach that addresses the reliability of SSS in the context of FPs. A literature review of existing service design approaches, service reliability characteristics and service modelling has been carried out. Utilizing the strength of computer software packages, a visualized, customized service design process has been developed and it will also facilitate service system optimization with respect to reliability.

The objectives of this thesis are to:

- Understand the basic ingredients of services;
- Clarify the definition of service reliability;
- Summarize the characteristics of service failures;
- Review the reliability assessment techniques with respect to their application to services;
- Review existing methodology for designing general services with specific reference to SSSs of FPs;
- Create innovative design approaches for SSSs;
- Analyze the reliability of SSSs through functional modelling;

- Find critical components of SSSs and give suggestions for optimizing SSSs with respect to their reliability.

1.3 Organization

The remaining chapters of this thesis are organized as follows:

Chapter 2 discusses the characteristics of service failures and service reliability. What a service is and what traits it has are explained, based on an in-depth literature review of service quality. The characteristics of service failures are summarized. General reliability assessment techniques are discussed with respect to their use in evaluating reliability of services. The chapter concludes with a discussion of the advantages and disadvantages of existing reliability techniques for service evaluations.

Chapter 3 introduces FPs and their advantages. The function and components of SSSs in the context of FPs are described, serving as the basic knowledge for service design. The state of the art of service design methodology is reviewed in chapter 4, where the design process of services is compared with that of 'hard' products.

Chapter 5 presents the principle of a 'bottom up' service design process, and a computer aided design tool is introduced. In comparison, a 'top down' service design process is proposed in chapter 6, where a Generic Service Support System (GSSS) developed to serve as a template in designing services is presented. Its effectiveness is validated through comparison with two real SSSs. Chapter 6 ends with a comparison between the 'bottom up' and 'top down' approaches.

Chapter 7 describes how a computational model for SSSs is built. Computational models of typical activities are presented. A trial of a small system is given.

Chapter 8 presents a case study of addressing reliability issues of the SSS for aircraft engines, where the 'top down' design approach is applied.

Chapter 9 concludes the thesis and gives directions for future research.

Chapter Two

Literature Review of Service Reliability

2.1 Introduction

Research into service reliability is in the pioneering days, although research into services has been carried out for decades. People try to understand what a service is and what constitutes service quality. This understanding is obscured by inherent qualities of 'service', such as intangibility, which radically differentiate it from hardware.

Gronroos (1988) identified reliability as one of six criteria in the assessment of service quality. However, little research has since been done on service reliability, although much effort has been made in service recovery strategies. This is because, in the general service industries, recovery actions tend to be effective and inexpensive to implement. For instance, the average recovery rate is about 8 on a scale of 1 (very poor) to 10 (very good) (Kelley *et al.*, 1993). Moreover, Colgate and Norris (2001) indicate that service recovery is not the unique reason a customer may stay with or leave a service organisation after a service failure. Other factors, such as loyalty and barriers to leaving, are just as prevalent in the decision-making process.

However, for some reliability-critical services, failure to provide the contracted level imposes rebarbative penalties upon service providers. For instance, the term C_o (cost of outages) is often used to represent the costs incurred by customers to power distributors when the utility is unable to meet their demand (Bums and Gross, 1990).

Nowadays, the rapid growth of complexity of modern engineering systems in which there are more integrated technology elements leads to a high cost of recovery. Also, in many engineering applications safety problems may occur if the support system fails to keep equipment operating in good order. For instance, in manufacturing industry, along with the success of FPs, an increasing number of manufacturers realize that a reliable SSS is pivotal for assuring the contracted availability of the core hardware product. Therefore, for engineering service support systems, the emphasis should be on creating reliable systems rather than on recovery strategies.

This chapter gives an introduction to what a service is and to its characteristics. A literature review of service failures in general service sectors has been carried out, based on which the characteristics of service failures are summarized. This is followed by an elaboration on the concept of service reliability and a comprehensive discussion of existing reliability assessment techniques with respect to service evaluation, whilst the advantages and disadvantages of these techniques are given.

2.2 Service

This section presents a brief introduction to the essence and characteristics of services, which serves as the jumping-off point for the discussion of characteristics of service failures and service reliability.

2.2.1 Definition of Service

Definitions of service are given in many works. The International Organization for Standardization (ISO, 1991) defines service as:

A subset of a product, a product being the result of a production *process.*

The definition according to Kotler (1983) is:

> *A service is any activity or benefit that one party can offer to another that is essentially intangible and does not result in the ownership of anything. Its production may or may not be tied to a physical product.*

Researchers also give some informal descriptions of services. Tortorella (2005) considers service as a function for use by a person or machine (a customer); Gronroos (1988) regarded services as largely intangible subjectively experienced processes. Realistically, services often take a variety of forms in different industries. Great differences exist between these various services, for instance, cleaning and security services, educational services, health care services and consultancy services (Edvardsson, 1992). In his words, service is not uniform in terms of the degree of standardization, labour and capital intensity, etc.

2.2.2 Service Characteristics

However, services share some common characteristics that differentiate them from conventional hardware products. Shostack (1984) stated that a service is not a physical object and can not be processed; in her words, 'services are unusual in that they have impact, but no form, just like light, and they can not be physically stored or processed and their consumption is often simultaneous with their production'. Further, she pointed out the difference between products and services stating that a product is defined by its existence in both space and time, and is tangible, whereas a service does not have a spatial element to it, and exists only in time. Gronroos (1988) summarized widely accepted characteristics of services as follows:

1) Services are more or less intangible;
2) Services are activities or a series of activities rather than things;
3) Services are to some extent produced and consumed simultaneously;
4) The customer participates in the production process at least to some extent.

These features not only differentiate services from conventional products; they also have profound influences on the evaluation of service reliability.

2.3 Service Failure

Due to these characteristics, service failures usually make themselves manifest in different ways from those of hardware products. Understanding service failure modes is a prerequisite to any study of service reliability. As claimed by Tortorella (2005), the peculiar failure modes of services are distinct from those which reliability engineers are accustomed to dealing with in tangible systems (such as electric generators, aircraft, radio transmitters). Therefore, identifying the characteristics of service failures is the departure point for measuring the gap in applying general reliability assessment techniques to service reliability evaluation.

This section explains types of service failure and presents characteristics of failures in 'high contact' service sectors like retailing and engineering support services (SSSs of FPs).

2.3.1 Service Failures in General Service Sectors

Service incidents have been collected by researchers using the Critical Incident Technique (CIT) (Flanagan, 1954). During the last decade incidents have been reported from general service sectors such as restaurants, retailing and airlines (Bitner *et al.*, 1990; Edvardsson, 1992; Kelley *et al.*, 1993; Chung *et al.*, 1998; Meuter *et al.*, 2000; Forbes *et al.*, 2005;). Nevertheless, the word 'service failure' is not often referred to, instead appearing such terms as 'service errors', 'unfavourable events' and 'dissatisfying incidents'. It is for the business and marketing investigation rather than the reliability research that the 'Critical Incidents Technique' (CIT) was adopted to collect the positive and negative incidents in these works. CIT is a data collection method mainly using interviews (see Flanagan, 1954).

There exist two types of service failure: contract failure and customer-perceived failure. Contract service failures occur when the service performance is worse than that specified in contracts. A review of the general failure modes of a wide range of service sectors reveals that the majority of service failures are contract failures. As reported by Edvardsson (1992), *Delayed* and *Cancelled Flights* account for 82 per cent of all the negative incidents in the airline industry. In restaurants, contract failures account for more than 70 per cent of all failures (Chung *et al.*, 1998). Table 2.1 lists typical contract failure modes in retail services.

Failure Modes	Descriptions
Slow or Unavailable Service	Stores did not service products that they sold, or store employees took too long to provide the service
System Pricing Failure	Individual-item pricing was not in agreement with the scanner-based price charged at the register
Packaging Errors	Wrong item was included in a package
Out of Stock	Products were advertised, but were not in stock

Table 2.1: Contract Service Failure Modes in Retailing (adapted from Kelley *et al.*, 1993)

In addition, customer perception is a key factor, and the evaluation of the severity of a service failure is more or less subjective (Kelley *et al.*, 1993; Chung and Hoffman, 1998). Customer-perceived failures are those which met contracted performance but did not meet customer expectations. Service failure can sometimes have a positive influence on the relationship between providers and customers if only the recovery process is perceived as good; that is, the customer feels valued by the service provider. Service failures can be recovered better than might be expected, but failures with bad recoveries tend to be memorable (Rotondaro and Olivera, 2001; Stefan, 2003). Human behaviour is, as would be expected, an important source of service system failures (Bitner *et al.*, 1990).

2.3.2 Service Failures in SSSs

2.3.2.1 Contract Failures and Customer-Perceived Failures

A contract for the SSS for a FP has Key Performance Indicators (KPIs) that define the level of service to be provided. The level of performance of each KPI may be monitored with time. Any fall below the contracted level of performance may be defined as a contract failure.

However, should the actual performance be well above the contracted performance for long periods, then customer expectations are raised. Any fall in actual performance below the expectation of performance is then perceived as a failure, even though the actual performance is not below the contracted performance. Though customer-perceived failures do not incur financial penalties on the total care product providers, they do harm the long-term relationship with customers.

2.3.2.2 Internal Failure and External Failure

Failures of such systems can alternatively be divided into two groups: external and internal failures. External failures are those that can be seen by customers and relate to service quality and customer expectations. Typical examples are low quality and late deliverables (contractual failure), and perfunctory and slow reaction to client enquiries (customer perceived failure).

Internal failures relate to system inefficiency and individual events are not evident to the customer. For example, the transfer of inaccurate data between activities may lead to repeat work and hence increased cost, but the customer may be unaware of the problem if the overall system performance is satisfactory.

2.4 Service Reliability

2.4.1 Definition of Service Reliability

Fitzsimmons (1998) stated that reliable service is a customer expectation and means that the service is accomplished on time, in the same manner, and without errors every time.

Service reliability is a new infant in the family of reliability engineering disciplines and is, to some extent an ambiguous concept. Tortorella (2005) observed that, in the telecom industry, network technicians often attribute 'network service reliability' to reliability of the hardware network. But actually it is, as he further explained, reliability of the network service carried on the hardware network. It is explicitly stated by Tortorella (2005) that service reliability is the reliability of the service carried on the service delivery infrastructure rather than the infrastructure's reliability. In other words, it is the reliability of a function, not of the physical infrastructure that delivers the function.

Tortorella (2005) defines service reliability in the context of the telecom industry as:

> *The capability of proper functioning of the service during the time it is offered by the service provider.*

He breaks service reliability down into three aspects:

1) Service Accessibility - the probability that a transaction can be initiated when desired;
2) Service Continuity - the probability that a transaction in progress is not interrupted or experiences no other type of poor quality before it is completed;
3) Service Release - a transaction, when completed, may be able to be dismissed.

Dai *et al.* (2003) implemented a study of the reliability of centralized heterogeneous distributed systems which have been increasingly applied in many safety-critical systems, such as the banking, military and nuclear systems. These systems are developed to provide different services with specific objectives such as running a computer program, controlling a production process, or completing some other task; therefore, the service reliability of the distributed system is a key point of the quality of service. Dai *et al.* (2003) defined service reliability of this sort of systems as:

> *The probability to successfully provide the service in a distributed environment.*

Alonso-Rasgado *et al* (2004) describe service reliability in the context of FPs and define it from a functional perspective as:

> *The probability that a service will generate defined functional outputs within prescribed limits, for a given period of time, using defined resources and under prescribed operational environment conditions.*

These definitions are given in light of different industrial backgrounds; therefore, brands exclusive to the specific service industry are involved. However, the emphasized key words including functioning, time and prescribed environment, are consistent with those of the classic reliability definition, although the nature of services incurs some special requirements for service reliability assessment.

2.4.2 Service Reliability Assessment

Service reliability is not the reliability of the infrastructure that delivers the service, but the reliability of the function delivered by the delivery infrastructure. It comprises not only hardware, but also 'soft' functional elements including people, data sources and decision-making processes. Even if the delivery infrastructure has all elements in good condition before servicing, it may still generate failed services as there are a lot of uncertainties during the process, for example, human behaviour. Therefore, service reliability assessments need to take into account factors from multiple disciplines.

Moreover, service reliability has to be evaluated over a period of time; that is, static analysis is inadequate to depict the reliability characteristics of a service. Since service only exists in time and is not storable, service reliability can not be assessed before or during its delivery and can only be assessed through the results of a bundle of similar services carried out over a span of time.

Traditional reliability techniques are usually 'bottom up' and they try to predict the reliability of an object through analyzing the components it comprises and the failure dependency between them. Unfortunately, although normally a service failure could be traced down to smaller events involved in the delivery processes, the casual relationship between the failure of components and that of a service system is not as explicit as that of hardware systems.

Classic reliability assessment models basically use historical observed performance data and current conditioning of objects to predict their reliability in future. However, services involve a large amount of human behaviour that is considered difficult to predict. Customers participate to some extent in the service production process; therefore, service providers can not fully control service reliability.

Furthermore, the existence of customer-perceived service failures makes it rather subjective for reliability assessments of services especially those 'high contact' services. Although not essential, it is desirable to consider customer-perceived failures because ignorance of them may be misleading when making corrective strategies. Evidence can be found in Edvardsson (1992) who interviewed both employees and customers and found their attitudes to ordinary problems in service offerings inconsistent. From the staff's viewpoint, contract failures like *delayed or damaged luggage* have top ranking, while customers deem *staff's bad attitude*, which is even not mentioned by staff, as one important source of service failure.

In contrast, reliability assessment of engineering support services is relatively objective as there is normally an agreement which specifies the contracted terms of the support services. For example, the AA (Automobile Association) explicitly specifies that their technicians have to arrive at the scene and start the vehicle breakdown recovery service within 40 minutes of receiving the call. Volvo Aero promises their clients that their SSS will not let a client's malfunctioning engine cause the aircraft to be halted on the ground for more than 24 hours.

2.5 Reliability Assessment Methodology

Characteristics of service reliability assessment create new requirements for reliability assessment techniques. This section reviews generally used techniques and presents a discussion of their advantages and disadvantages when applied to service reliability evaluation.

2.5.1 Reliability Block Diagrams (RBD)

The Reliability Block Diagram (RBD), the first formal reliability assessment model, uses a number of interconnected functional blocks to describe the effect of each block failure on overall system reliability. It is a representation of the effects on relationships between failed components rather than on the physical connections in the real system. Basically, the RBD depicts the relationship in two typical ways: series and parallel.

Series connections imply that all the elements must work all of the time for the system to operate, and the reliability of the system $\boldsymbol{R_s}$ is normally calculated as:

$$R_s = R_1 \times R_2 \times R_3 ... \times R_n \tag{2.1}$$

In comparison, parallel connections imply a redundancy mechanism; that is, the system will continue to function satisfactorily provided that any single element of the system works well. For such a system to fail all elements must fail. The system reliability $\boldsymbol{R_s}$ can be calculated as (Thompson, 1999):

$$R_s = 1-(1-R_1)\times(1-R_2)\times(1-R_3)\times\ldots(1-R_n) \qquad (2.2)$$

Many variations are deduced including the *active redundancy, partial active redundancy*, *standby* or *inactive-parallel* and *K-out of-N* systems. The advantage of RBD is that it presents the causal dependency of failures in an intuitive manner. In hardware systems, the failure state of components may often be defined precisely and, if the failure rates of particular components are known accurately, then the overall reliability of a system may be calculated. Redundancy, both active and stand-by, may be included in such calculations.

With respect to service reliability evaluation, RBD is hardly applicable because such modelling is not generally possible in the case of services. A particular activity can rarely be defined as 'failed'. Rather, it can be seen to slow down, say because there are fewer people working that day, or there may be a delay in the supply of some piece of equipment. It may be that the wrong information (or material) is supplied and that a particular activity may need to be repeated.

In addition, RBD is unable to depict the causal relationship between activity failure and service system failure because such a relationship is not so explicit in service systems. In a complex service system, individual events such as repeat or tardy work may not cause a failure of the whole system in the sense that the total system performance does not fall below requirements. Repetition of the failed activity only leads to an increase of the total service time consumed but the service does not necessarily fail as long as the consumed time is still within the prescribed range.

2.5.2 Fault Tree Analysis (FTA)

Fault Tree Analysis (FTA) is a deductive method which proposes the question, 'if this happens, what can be causes'. It depicts the causal relationships leading to a specific system failure (top event) by constructing a logical tree of basic events using symbols such as AND or OR gates. An advantage of FTA is that in contrast to the RBD, the basic events need not necessarily be component failures, which makes it applicable for the service designer to identify potential reliability threats including human and other non-hardware failure causes.

When delivering a service, activity failures (internal failures) do not necessarily cause the service to fail; however, if there is an unfortunate combination of adverse individual events, then a system failure event may be created, which is an 'external failure'. This may be evident by not achieving KPIs, such as response time to an incident, or the delivery of a remanufactured product on time. FTA could be utilized to identify such possible combinations.

The main disadvantage of FTA is that it only represents the static causal relationships. As service reliability concerns time, it demands a dynamic evaluation approach to model the repetitions incurred by activity failures. In addition, the hazard rate information of basic events such as activity failure and inappropriate human behaviour are rarely available; therefore, in service reliability evaluations, FTA could be used more as a qualitative analytical tool rather than for quantitative reliability evaluations.

2.5.3 Markov Model

The Markov Model is normally used to cope with systems subjected to a repair strategy. For a repairable system with finite possible states, the Markov Model proceeds by the enumeration of system states and uses the transfer rates to represent the probability for the system to stay in each state. The simplest situation is a system comprising only two states: work and fail, and the failure rate and repair rate are constant; see figure 2.1.

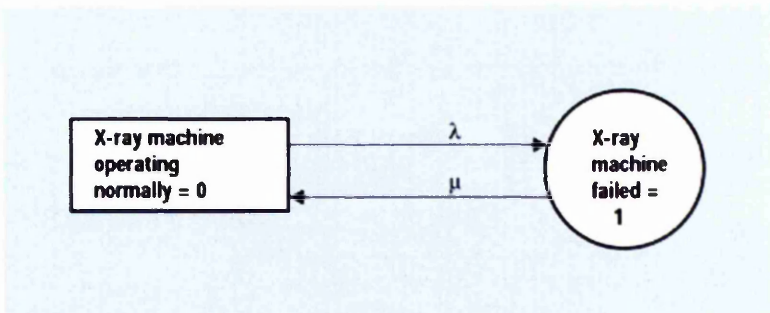


Figure 2.1: X-ray machine transition diagram (*Source: Dhillon, 2001*)

Many implementations of the Markov Model are described in the literature. For example, Akhtar (1994) applied it in evaluating the imperfect fault-coverage problem; Walker *et al.* (1989) utilized it to model the effects brought by a redundancy management policy. The Markov Model is often used in evaluating the reliability of multi-state systems.

Research applying Markov Models to study the reliability of service systems has been done. For example, Dai *et al.* (2003) model the reliability of a centralized heterogeneous distributed system that provides reliability critical services through using the Markov Chain to model the reliability of three virtual servers in redundancy and the reliability of each node on the route of delivering the service. Another example is Huang and Billinton (2007) who use the Markov theory to model the reliability of an electric power service system through investigating the possibility that the system remains in a good condition. Such service reliability analysis assumes that the contracted service can always be delivered as long as the service delivery infrastructure is in a good condition. That may be mostly true in service systems that do not involve much human behaviour; however, such analysis is not applicable in SSSs which involve a large number of human activities to deliver services.

SSSs could be defined as being in the failed state when they function poorly; that is, the service mission is not delivered at the contracted level. Another service mission could be carried out to amend the service mistakes and pull the service performance back to above the contracted level. Such single service missions are not repairable as they only exist in time and are consumed simultaneously with delivery. The problem of applying the Markov Model to assess the residence time of a SSS above the contracted level is that the failure rate of the SSS and its repair rate are rarely available.

2.5.4 Failure Mode and Effect Analysis (FMEA)

Failure Mode and Effect Analysis (FMEA) is a risk assessment technique for systematically identifying potential failures in a system or a process. It is widely used in manufacturing industry in various phases of the product life cycle. In FMEA, failures are prioritized according to how serious their consequences are, how frequently they occur and how easily they can be detected. An FMEA also documents current knowledge and actions about the risks of failures, for use in continuous improvement.

The purpose of the FMEA is to take action to eliminate or reduce failures, starting with the highest-priority ones. It may be used to evaluate risk management priorities for mitigating known threat-vulnerabilities. FMEA helps select remedial actions that reduce cumulative impacts of life-cycle consequences (risks) from a systems failure (fault).

Rotondaro and Olivera (2001) show the use of FMEA in a case study to identify and mitigate potential critical failures in a medical clinic restaurant. A simple service blueprint is developed and 'front office' and 'backroom' in such a high contact service sector are determined. Failure modes perceived in 'moments of truth' are then revealed by interviews with customers. In contrast to the original FMEA, the consequences of such failures are evaluated by four criteria and associated risk indices:

1) Severity (S)
2) Likelihood of occurrence (O) (also often known as probability (P))
3) Inability of controls to detect it (D)
4) Recuperation (R)

Recuperation is the capacity to correct the service before or as soon as the customer perceives the failure, in such a way that the customer agrees with the correct action. Rotondaro and Olivera (2001) give the reason for adding such a parameter, that is, it is possible that the actual process delivering the service itself corrects the failure as it occurs in the presence of the customer, minimizing its effect. Thus, the FMEA scheme in this case has six indices ranging from 1 (lowest risk) to 5 (highest risk), as shown in table 2.2.

Index	**Failure Modes**
1	Badly-Set Tables
2	Impolite Employee
3	Tableware Missing
4	Cold Meal
5	Cleaning and Food Hygiene
6	Delay

Table 2.2: Six Main Failure Modes (adapt from Rotondaro and Olivera, 2001)

	Group 1					Group 2					Group 3				
	O	S	D	R	RPN	O	S	D	R	RPN	O	S	D	R	RPN
1	5	3	2	5	150	1	3	1	2	6	2	3	2	3	36
2	5	3	3	5	225	2	3	2	4	48	1	2	3	2	12
3	3	2	3	4	72	4	3	2	3	72	3	3	2	3	54
4	5	3	1	3	54	1	2	1	3	6	3	3	2	5	90
5	3	5	1	4	60	2	5	2	2	40	2	4	2	3	48
6	4	4	2	3	96	5	2	2	3	60	4	3	2	3	72

Table 2.3: Reference Values for the Three Groups (adapted from Rotondaro and Olivera, 2001)

Three different areas of the restaurant were studied independently by three well trained investigators. Results show that different Risk Priority Numbers (RPN) are assigned to the same failure modes; see table 2.3. An important cause given by Rotondaro and Olivera (2001) is that compared to the well defined failure modes and effects of products, the subjectivity of investigators dominates the way they perceive how the customers are reacting.

In order to detect a significant discrepancy among the groups, the mean and the range of scores for each group are used. Most subjective items and those presenting more homogeneous results can be observed. Although corresponding preventive actions are proposed and improvements in the service are demonstrated in the case study, it reveals the main limitation of using FMEA to evaluate service reliability, that is, compared to well defined failures and effects of products, effective evaluation of service failures often involves the inevitable subjectivity.

2.5.5 Monte Carlo Simulation

A Monte Carlo method is a technique that involves using random numbers and probability to solve problems. The term Monte Carlo Method was coined in reference to games of chance, a popular attraction in Monte Carlo, Monaco

(Wittwer, 2004). Monte Carlo Simulation is distinguished from deterministic reliability assessment techniques by being stochastic, that is by using random numbers (in practice, pseudo-random numbers) as inputs, which makes it especially suitable for simulating a service system that needs randomly generated values for uncertain activity performances over and over. Such performances have a known range of values but need an uncertain value for any particular time analysis.

It is also possible to carry out sensitivity analysis using simulation through analyzing the effect of varying inputs on outputs of the modelled system. More often, because of the repetition of algorithms and the large number of calculations involved, Monte Carlo simulation needs to be facilitated by using a computer.

2.6 Summary

This chapter introduced the peculiarities of services as opposed to traditional products. As a result of these peculiarities, failure of services manifests different characteristics. In the general service sector, service failures have been well documented and are often considered as measures of service quality. Contract failures, delayed flights, account for the majority of failures but customer-perceived failure is also important. Recovery strategies in the event of a failure are particularly significant, because recovery actions tend to be cost effective and, if good, can leave a very favourable impression in the mind of the customer. However, the recovery of engineering support systems involving costly hardware is less cost effective due to the financial penalties in contracts. Also, in engineering systems, certain failures cannot be tolerated for safety reasons.

Reliability is a key factor in the performance of service support systems because the aim is to provide a guaranteed level of performance over a long period of time. Contract and customer perceived failures are important. Contract failures are when the system performance with respect to defined KPIs falls below specified levels. Customer-perceived failure can occur if performance falls below expectations even though contracted performance has been achieved. For example, if a high level of performance has been routinely given then that may well become the reference rather than the contracted KPI.

Failures of SSSs can be external or internal. External failures are those seen by the customer. They occur when a number of individual service activities are poor and their combined effect reduces the overall system performance below contract, or perceived, levels. Internal failures are not visible to the client. They increase the time taken to provide the service, and they increase cost, but they may not have a total effect on the system KPI such that it falls below the contract (or customer-perceived) level.

Despite the plethora of research on hardware reliability, service reliability remains among the least studied and understood topics. Research into service reliability has just begun and no general analysis method has yet been set up. Classic reliability assessment techniques have limitations in evaluating the reliability of services. However, the functional reliability of a service system may be investigated through simulating a system network model which takes into account the uncertainties existing in the service system.

Chapter Three

Basic Elements of Service Support Systems

3.1 Functional Products (FPs)

3.1.1 Description of Functional Products

In FP provision, hardware equipment is not traded in the conventional manner but is incorporated in integrated SSSs. FPs bring a degree of flexibility of purchase and do not necessarily lead to the transfer of ownership of the hardware product. Customers may purchase hardware plus support services that enable the customer to benefit from a total functional provision. Alternatively, suppliers may retain the ownership of the hardware while providing support services; or provide only support services for customer-owned hardware. In all cases, customers will be given a guarantee of a certain level of availability of the hardware, which calls for highly reliable SSSs to keep it operable. Such services involve more functions and a higher level of complexity than does the after service of general products such as laptops.

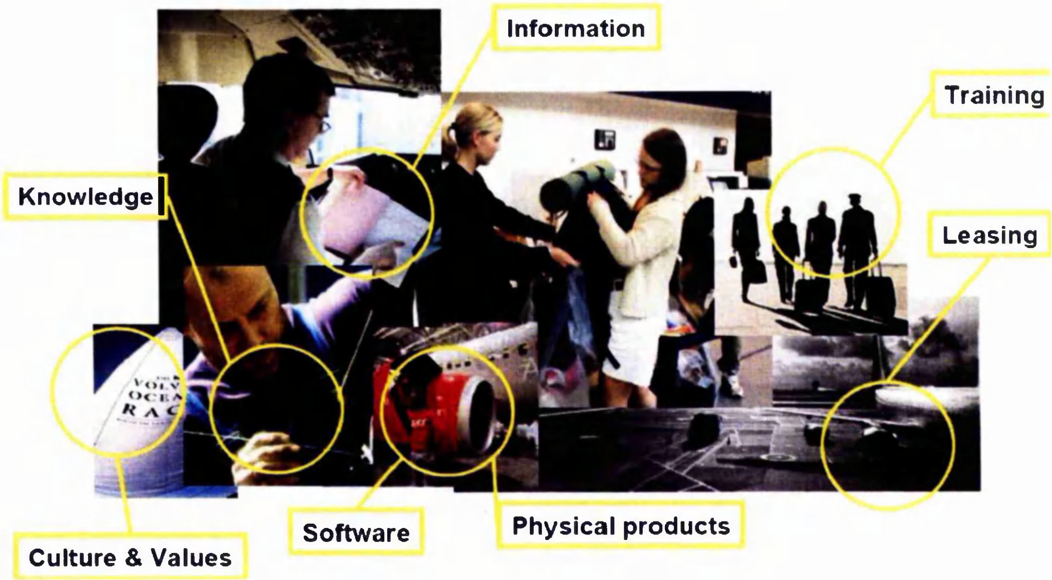


Figure 3.1: Concept Illustration of FPs (source: www.volvo.com)

FPs consist of many heterogeneous elements. For example, a functional aircraft engine comprises the engine itself (mechanical and electronic components) and a SSS which comprises functional elements of people, data sources and decision-making processes; see figure 3.1.

3.1.2 Paradigms

To depict a whole picture of FPs and illustrate how the innovative provisions are achieved, three paradigms from the current market are introduced:

1. TotalCare® Aircraft Engine

A well known example of FPs is the TotalCare® aircraft engines provided by Rolls-Royce. Clients pay only for the use of the engine and do not purchase the hardware. Meanwhile a SSS run by Rolls-Royce is responsible for keeping the engine to the contracted level of availability.

Against an agreed rate per flying hour fee, TotalCare® offers the opportunity to remove uncertainties from engine management and provides greater financial confidence from managing predictable costs. Charles Cuddington, managing director of Rolls-Royce, believes that these comprehensive aftercare packages are able to deliver a range benefits, including predictable budgeting for clients (Travel & Tourism 2003).

In a similar manner, Volvo Aero Corporation (VAC), one of Rolls-Royce joint-venturers on the new Trent 900 project, provides functional aircraft engines for almost all the jet engines for the Swedish Air Force, including the Gripen fighter.

As problems of aircraft engines are usually safety-critical and highly complex, it takes weeks to implement comprehensive inspection and repair. Therefore, to achieve the contracted level of an engine's functional availability, overhauls are almost entirely done on an exchange basis rather than in the repair process.

2. Fully Supported Machinery

This is an example of the supply and support of key process machinery that performs a defined function as part of a production process, rather than the sale of the equipment.

To maintain all Dividers and Moulders provided to Allied Bakeries in a good condition, APV Baker, the supplier of such FPs, tailored a set of comprehensive support services. Dedicated APV Baker engineers maintain the equipment located at all Allied Bakeries sites, against predetermined maintenance checklists; they also carry out repairs in the event of a breakdown (see figure 3.2). The engineers visit each bakery in the participating group on a regular basis, according to a previously agreed schedule.

Figure 3.2: The Fully Supported Machinery (source: www.apv.com)

The agreements between APV Baker and the Allied Bakeries specify the annual schedule of engineers' work. All urgent works are required to be completed prior to holidays, and working engineers should cover any emergencies that may arise whilst other dedicated engineers are on holiday.

3. Total Landing Gear Support

Total Landing Gear Support (TLS), launched by Lufthansa Technik, is a cradle-to-grave landing gear support system. That is, under TLS, Lufthansa Technik assumes total responsibility for a customer's landing gear from the moment the aircraft is purchased, including monitoring time in operation, Aircraft on Ground (AOG) support, overhaul, exchanging gear, leasing and loans, right through to resale; see figure 3.3.

Figure 3.3: The Total Landing Gear Support (source: www.lufthansa-technik.com)

As with TotalCare® aircraft engines, landing gear overhauls are almost entirely done on an exchange basis rather than in the closed-loop process. For example, the landing gear overhaul for a Boeing 747 takes five to six weeks, while three to four weeks have to be allowed for the landing gear of a narrow body aircraft. Lufthansa Technik clients receive an overhauled landing gear assembly at the agreed time and the unserviced unit is then sent back for overhaul. The overhauled gear is then ready for the next customer.

Lufthansa Technik believes that it can save money and pass these savings on to the customer in a number of ways, including:

- With complete control over customers' landing gear, Lufthansa Technik's landing gear specialists should be better able to monitor, maintain and schedule overhauls to reduce the cost of overhauls.

- Lufthansa Technik currently keeps 197 exchange legs on hand. Providing needed spare landing gear for all of its customers from its large, portfolio of spare landing gear means that customers themselves don't have to keep costly spare landing gear in stock.

- Third, the airline can achieve quality control, economies of scale, and economies of specialization by maintaining four large, specialized, high-volume, overhaul centres with efficient industrial operations.

Through improved maintenance processes and cooperation with the line maintenance department, Lufthansa Technik attempts to control costs by reducing the need for such costly repairs. As described in its website, already during 2006 and in recognition of the peculiarities of landing gear overhauls, Lufthansa Technik has set up a dedicated Landing Gear Services product division with four locations around the world. With its facilities in Hamburg, Hawker Pacific Aerospace in Los Angeles and London and Ameco Beijing, Lufthansa Technik Group is now ideally placed to provide the global market with services pertaining to any conceivable aspect of landing gear maintenance, repair and overhaul (www.lufthansa-technik.com).

3.1.3 Advantages of Functional Products

For customers, guaranteed availability of the FPs they purchased brings reduced risk as the SSSs ensure peace of mind for the overall lifetime support of the engine, from the time the engine is delivered to the customer until the engine goes out of service. This feature enables the operator to concentrate on core business. Moreover, not having to pay for the expensive equipment reduces capital expenditure on new projects. The advantages of FPs to customers can be summarized as:

- A high capital investment is not required; instead, there is a predictable periodic expenditure.
- Risk management; a guaranteed level of availability is provided.
- Technologically advanced equipment is guaranteed; therefore equipment does not become dated.

On the other hand, FP suppliers benefit from establishing a long-term business-business relationship with clients, which leads to long-term, stable revenue and greater knowledge of operating conditions. The advantages of FPs to suppliers can be summarized as:

- A steady state cash flow over the life of the FPs.
- The generation and promotion of a long-term relationship with the customer and provision of good feedback to facilitate hardware and service improvement.
- The benefit of sustainable design plus competitive advantage.
- Increased knowledge of the product in the working environment; this in turn will lead to reduced product failures and increased availability.

3.2 Service Support System (SSS)

Figure 3.4 shows some key elements of a SSS. Such systems are multi-disciplinary. Hill (2002) describes a wide range of opportunities for improved service design, including service design for manufacturing. The introduction of new technologies and the development of integrated supply chains are seen as factors that increase the importance of providing effective services. The services add value to the customer's use of the tangible product and lower the total life-cycle cost of the product.

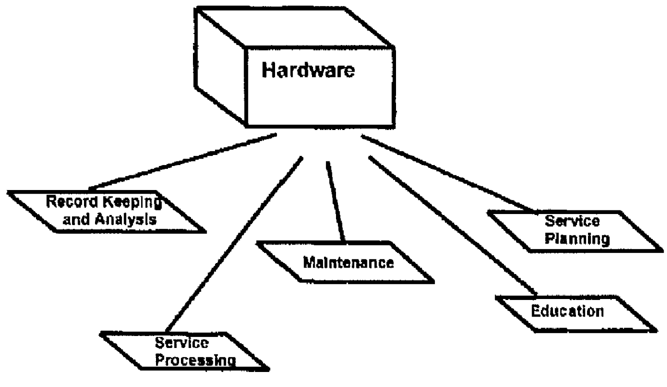


Figure 3.4: Elements of a total functional product (source: Alonso-Rasgado *et al.*, 2004)

In FP provision, the availability of hardware relies on the efficiency of the SSS. Moreover, minimizing the down time of a 'Total Care' engine leads to an increase in profit. The service delivered by the SSS is more than maintenance, although this is actually one of its main functions; it may include processes of decision-making and operations planning, remanufacture and education.

The images of such systems are ambiguous as they inherently contain a mix of tangible and intangible elements. The system can be understood from two perspectives. One is to view it through the physical perspective in which all tangible and intangible elements are categorized into different groups according to the role they play. Alternatively, it is possible to view the systems as a set of processes and activities working successively to deliver the function to customers. In the following two sections, these two cases will be introduced sequentially.

3.2.1 Physical Structure of SSSs

Although services are not uniform and they often present different modalities in various industries, service systems comprise some common components that may be tangible or intangible. Bitran and Pedrosa (1998) provide an insightful illustration of components of service systems and classify them into three groups shown as shown in figure 3.5.

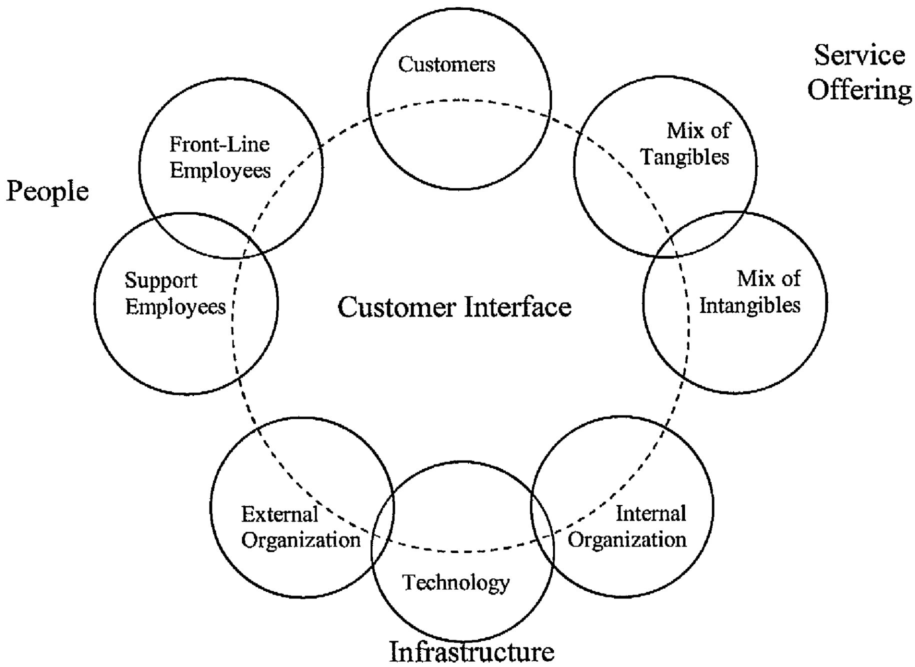


Figure 3.5: Components of Service Systems (adapted from Bitran and Pedrosa, 1998)

Three groups, people, infrastructure and service offering, were further divided as Customers, Support Employee, Front-line Employee, Internal Organization, External Organization, Technology (Soft and Hard), Mix of Tangibles such as core products and service evidence, and Mix of Intangibles such as environment, entertaining and ambience.

One special component should be highlighted, that is, service evidence which plays the crucial role of verifying either the existence or completion of a service. According to Shostack (1981), this evidence may be peripheral, that is possessed as part of the purchase, but of little worth independently; or essential, not possessed by the customer but still having 'a dominant impact on service purchase.

Understanding the service systems in this manner brings two design handicaps. One is, as Bitran and Pedrosa (1998) claimed, that the distinction between tangibles and intangibles is blurred. The other is the existence of human factors, which are not as predictable as products. Moreover, system design aims to create logical, formal models to represent function rather than to create a physical system, that is, it does not involve the selection of hardware or detailed specification of software (Harrison and Petty, 2002). Therefore, it is suggested the system should be analyzed from a functional perspective.

3.2.2 Functional Structure of SSSs

As claimed by Bitran and Pedrosa (1998), it is sometimes possible to describe the service system in different ways (e.g. through functions and processes) that may provide different insights and enrich the development effort.

There seems to be a consensus around the belief that services are processes and activities (Shostack, 1981; Lovelock, 1984). Just as Shostack (1981) pointed out, services consist solely of acts or processes and exist in time only. For functional products, the service (service support system) is the set of functions or activities that enable the hardware to be integrated into a total functional provision.

Generally, a SSS is extremely complicated. Figure 3.6 shows the functional level (top level) description of a SSS for functional aircraft engines. The inputs of the system include data, hardware and resources and the output is a functional engine. Each activity may be defined by a process diagram.

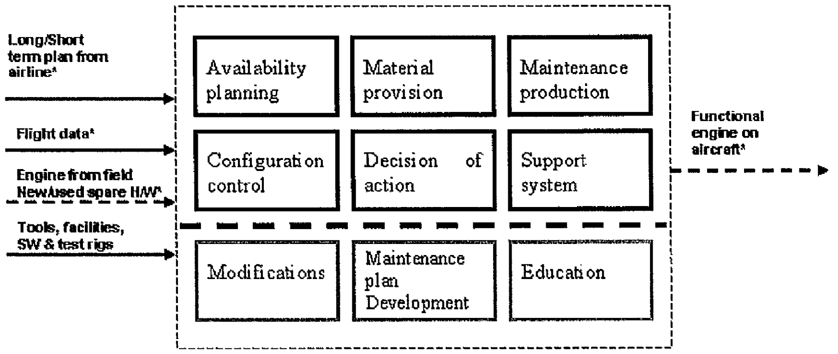


Figure 3.6: Functional level of SSS (adapted from Persson, 2004)

The functional or top level can be extended to sub-functions until the activity level is reached. Systems are divided into higher-order systems (2nd order, 3rd order). The principal functions (figure 3.6) are constituted by particular activities which are connected by their respective inputs and outputs in a system network; see figure 3.7.

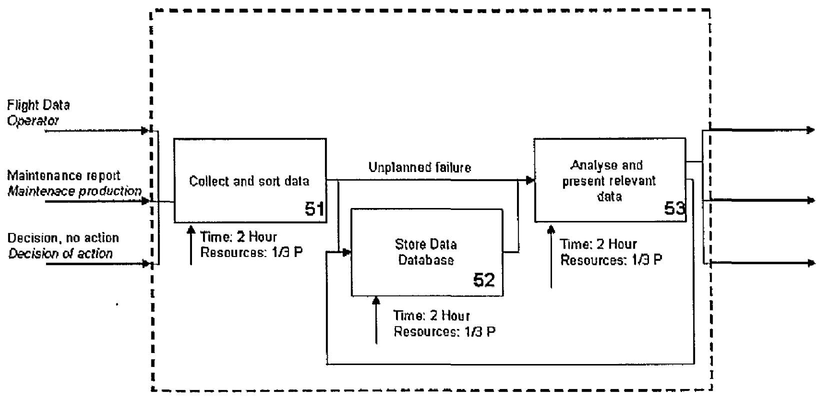


Figure 3.7: Activity level (source: Persson, 2004)

3.3 Typical Functional Components of SSSs

3.3.1 Activity with Single Input and Output

Figure 3.8: Activity with Single Input and Output

This sort includes all activities that require unique input; see figure 3.8. Normally the input comes from a fixed source and the output is sent to a fixed destination. Activities of this sort are usually not complicated; for example, *collecting information from clients* is triggered by a client's service request and the output is systematically recorded customer information, as shown in figure 3.9.

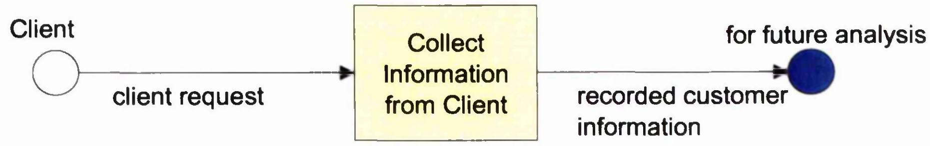


Figure 3.9: Collecting Information from Clients: example of activity with single input and output

3.3.2 Activity with Multiple Inputs

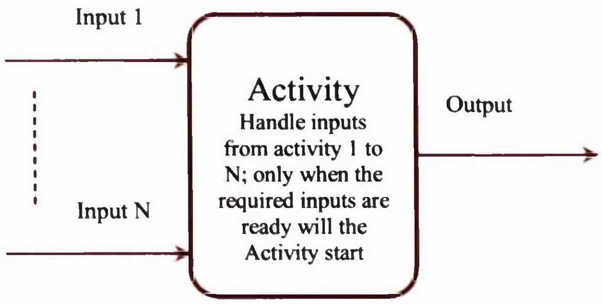


Figure 3.10: Activity with Multiple Inputs

Some activities have more than one input; to carry out an activity, all required inputs must be ready (see figure 3.10). This case could be further divided into two sub-groups:

1. Activity with Multiple Mandatory Inputs

In this case, all the inputs 1 to N must be ready to trigger the implementation of activities. One example is *Arrival Inspection* for verifying engine problems; in figure 3.11, failed hardware from client, and a technician, must be ready.

Figure 3.11: Arrival Inspection: example of activity with multiple mandatory inputs

2. Activity with Multiple Selected Inputs

Unlike the activity requiring multiple mandatory inputs, this kind of activity only needs one or one combination of the several inputs as trigger. An example in real industry is *Repair and Assemble*. To trigger the activity, either verified failure and spares or the failed testing result is sufficient; see figure 3.12.

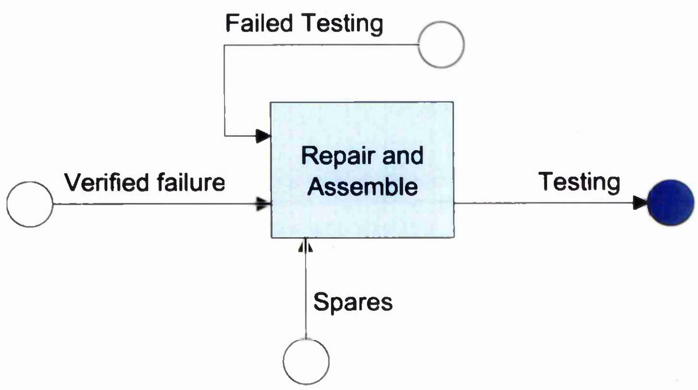


Figure 3.12: Repair and Assemble: example of activity with multiple selected inputs

3.3.3 Activity with Multiple Outputs

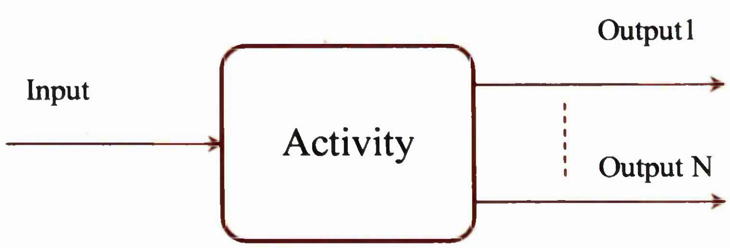


Figure 3.13: Activity with Multiple Outputs

Sometimes, activities do not output a unique result. As with an Activity with Multiple Inputs, two sub-groups could be further classified:

1. Activity with Exclusive Outputs

In this case, the activity outputs one of many mutually exclusive possible results depending on inputs and the system's conditions. Activities need to judge the content of inputs. One example is deciding whether maintenance should be off-site or on-site (figure 3.14).

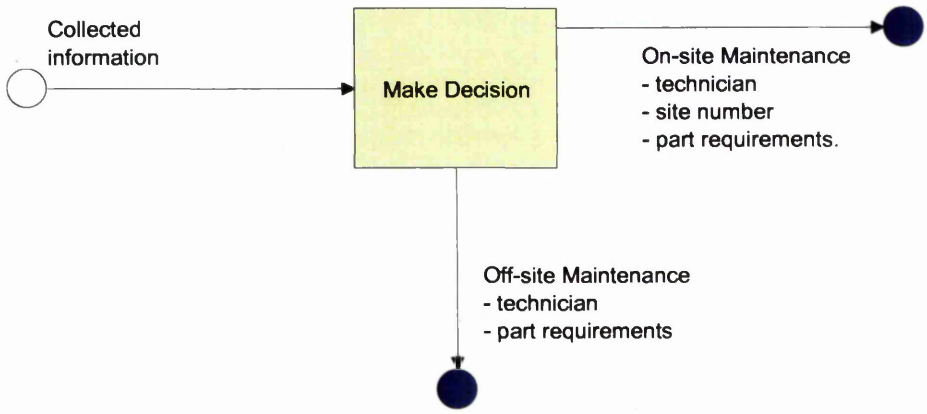


Figure 3.14: Make Decision: example of activity with exclusive outputs

2. Activity with Parallel Outputs

In contrast, an activity may output parallel multiple pieces of information and send them to different destinations. One example is generating a service report which sends certificates to clients and a report to an internal database.

More often, there exist hybrids of the two cases. An example is analyzing a customer's request (figure 3.15). Given one request from the client, the analysis result may be advice (instructions), the hardware order or service requirement. In the last case, client information will be sent to the database as well.

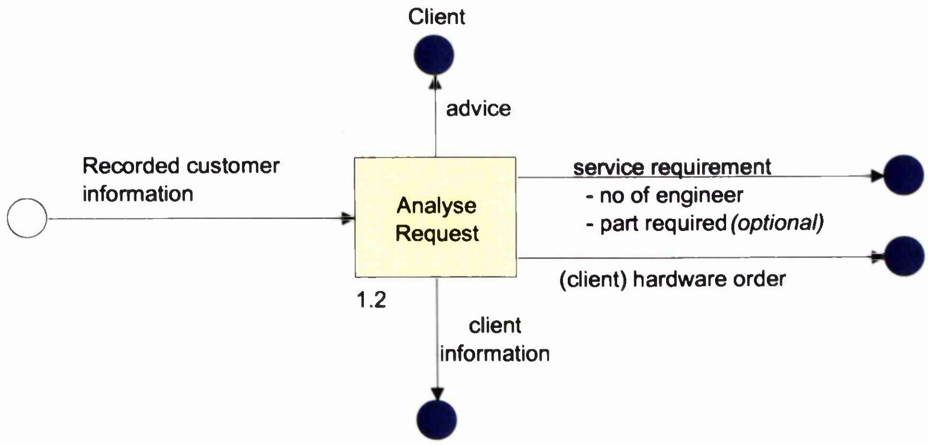


Figure 3.15: Analyse Request:: example of activity with parallel outputs

3.3.4 Activity with Multiple Inputs and Outputs

More often, activities have hybrid of multiple inputs and multiple outputs (figure 3.16).

Figure 3.16: Activity with Multiple Inputs and Outputs

An example is *Check Stock Level,* where inputs could be either the request from the client for hardware spares and hardware availability information, or the request from technicians together with the hardware availability information. The outputs could be the information of spare specifications to dispatch if the requested spare is in-stock, or an order request if the spare is out of stock (figure 3.17).

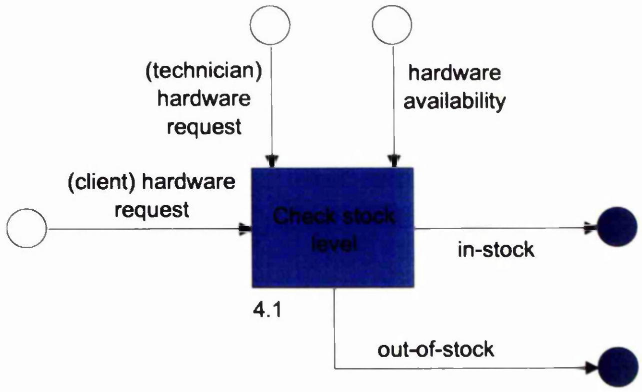


Figure 3.17: Check Stock level: example of activity with multiple inputs and multiple outputs

3.4 Summary

FPs comprise hardware and SSSs, and are becoming increasingly popular as manufacturers see them as a way to gain a competitive edge. Moreover, FPs help lower customers' expenditure, reduce business risk and continuously update technology.

SSSs aim to keep the core equipment in operating condition. Such systems are multi-disciplinary and contain mixed tangible and intangible elements which could be viewed from both the physical and the functional perspective. In the latter view, a SSS is a set of activities which together produce and deliver the contracted function. Typical functional elements can be seen through the number of their inputs and outputs, based on which four basic kinds of components are categorized.

Chapter Four

Literature Review of Service Design

4.1 State of the Art in Service Design

With the flourishing of the service industry, comes intense competition. Over 20 years ago, it was declared by Shostack (1984a) that as companies endeavour furiously to enter into the service economy, only those that gain control of the design and management process will survive and prosper.

Well designed service can bring not only customer satisfaction but also direct, indirect and incidental financial benefits. This can be observed from a story narrated by Hollins (2006): a hospital with an overcrowded car park was about to purchase more land to increase the size. After the design of its outpatients department was improved through a service blueprint exercise, it was found that through making the service far more efficient, the customer experience improved and the throughput of patients increased; in addition, the car park was significantly less crowded and the purchase of land to enlarge it was unnecessary, saving £1million. However, realistically, conscious service design is seldom implemented. As argued by Menor *et al.* (2002), the generally accepted principle behind new service development (NSD) was that 'new services happen' rather than evolving through formal development processes. Therefore, Hollins (2006) considers that it is comparatively easy for a company to gain a competitive advantage through the application of some quite simple design techniques.

Fortunately, the significance of service design has been recognized albeit only recently. Unfortunately, there are few developed tools to facilitate this effort. As commented by Hollins (2006), as soon as managers involved in service organizations realized that a conscious effort in applying 'design' techniques to service could lead to greater customer satisfaction, greater control over their offering and greater profits, they found that few resources were available to assist the application of design to their service products.

Due to their high level of complexity, SSSs of FPs demand a set of techniques to achieve and optimize reliable, economic but efficient performance. After the functional components of SSSs have been illustrated, this chapter examines how to integrate them feasibly and efficiently under prescribed design specifications. A comprehensive review of the literature on service design is presented in this chapter. Historical research into the sequence and steps involved in designing a service system is expounded first. This is followed by further illustration of the main activities and targets of each step, after which the design of SSSs is examined in more detail. Finally, advantages and disadvantages of existing methodology are discussed.

4.2 Service Design Processes and Activities

4.2.1 General Processes of Service Design

The literature reveals a consensus that designing services follows similar processes to designing hardware products (Bitran and Pedrosa, 1998; Alonso-Rasgado and Thompson, 2006). Alonso-Rasgado and Thompson (2006) divided the service design process into three distinct stages, each of which has equivalent steps in the hardware design process; see figure 4.1.

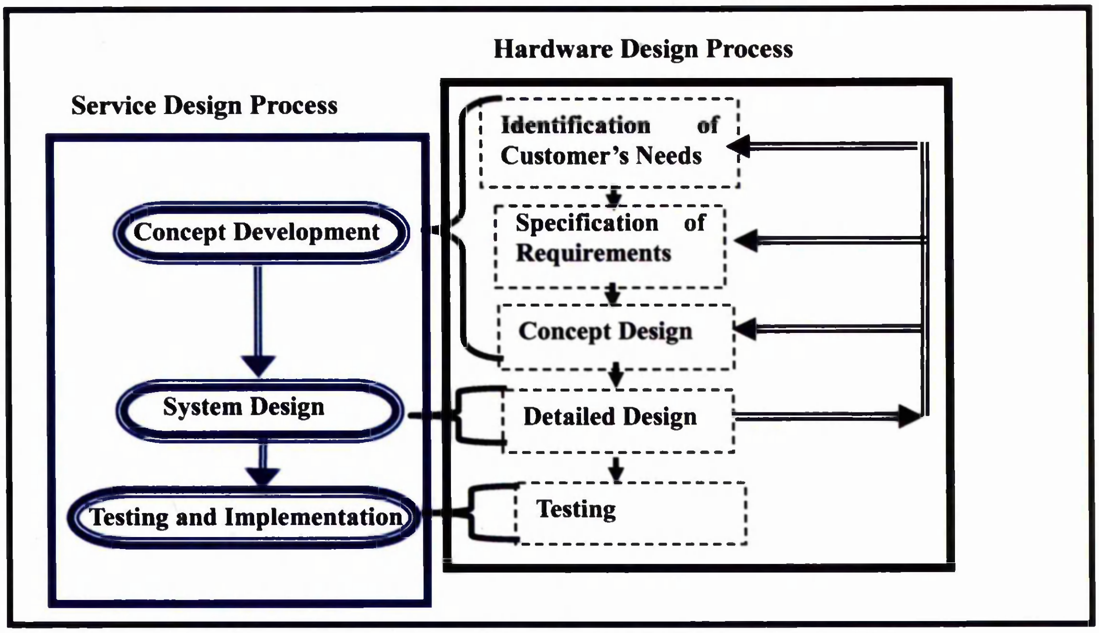


Figure 4.1: Comparison of service and hardware design processes (source: Alonso-Rasgado and Thompson, 2006)

Both hardware and service design processes start by obtaining an understanding of the customer's needs, which is carried out in the Concept Development Stage. However, according to Bitran and Pedrosa (1998), service concept development requires special attention to intangibles such as ambience or social needs. The second stage in the process is System Design, where specific design details of the service are produced. The final stage consists of testing and implementation of the service designed.

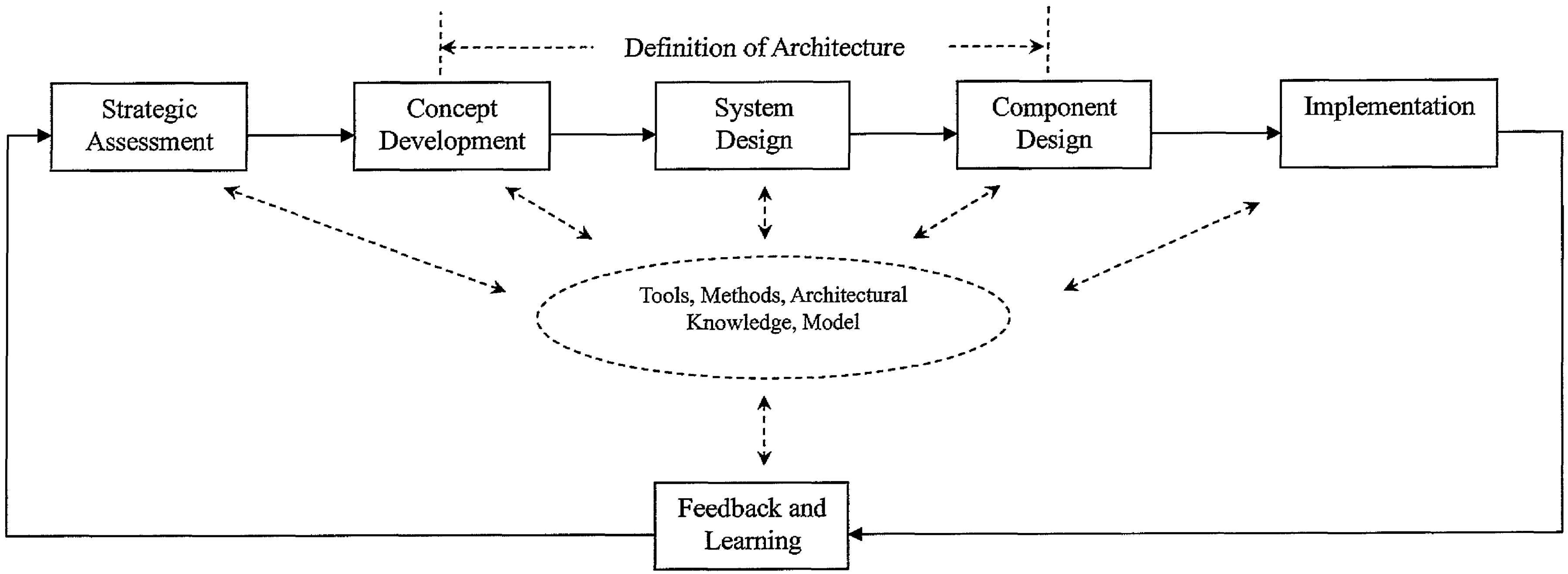


Figure 4.2: General Sequence of Stages of Development Projects (source: Bitran and Pedrosa, 1998)

Figure 4.2 shows the general sequence of stages in project development. As stated by Bitran and Pedrosa (1998), during each stage, knowledge will be added to enrich the design. Differing slightly different from the processes shown in figure 4.1, the development commences with strategic analysis which clarifies commercial goals from the service provider's perspective. In the context of FPs, a SSS is integrated with hardware as a whole package; the strategic assessment is stressed when considering the whole package offering. The component design is further separated from the system design stage and system testing is integrated with implementation. Feedback and learning is specially highlighted and deemed to be significant in helping service providers update awareness about customer requirements.

The specific development model for services can be found in Scheuing and Johnson (1989a), as in figure 4.3; this decomposes the linear service design process into continuous activities and shows the involvement of customer-contact staff and customers in the process. In this model, people are grouped into three categories: users, customer-contact staff and operational staff, which are consistent with the categorization made by Bitran and Pedrosa (1998) who consider that customers, front-line employees and support employees are the people involved in services.

This model was further adapted in Menor *et al.* (2002), shown in figure 4.4, where the service design process is divided into four stages, but the sequence of main activities is similar. It can be observed that Menor *et al.* (2002) preferred to depict service design as a closed-loop process.

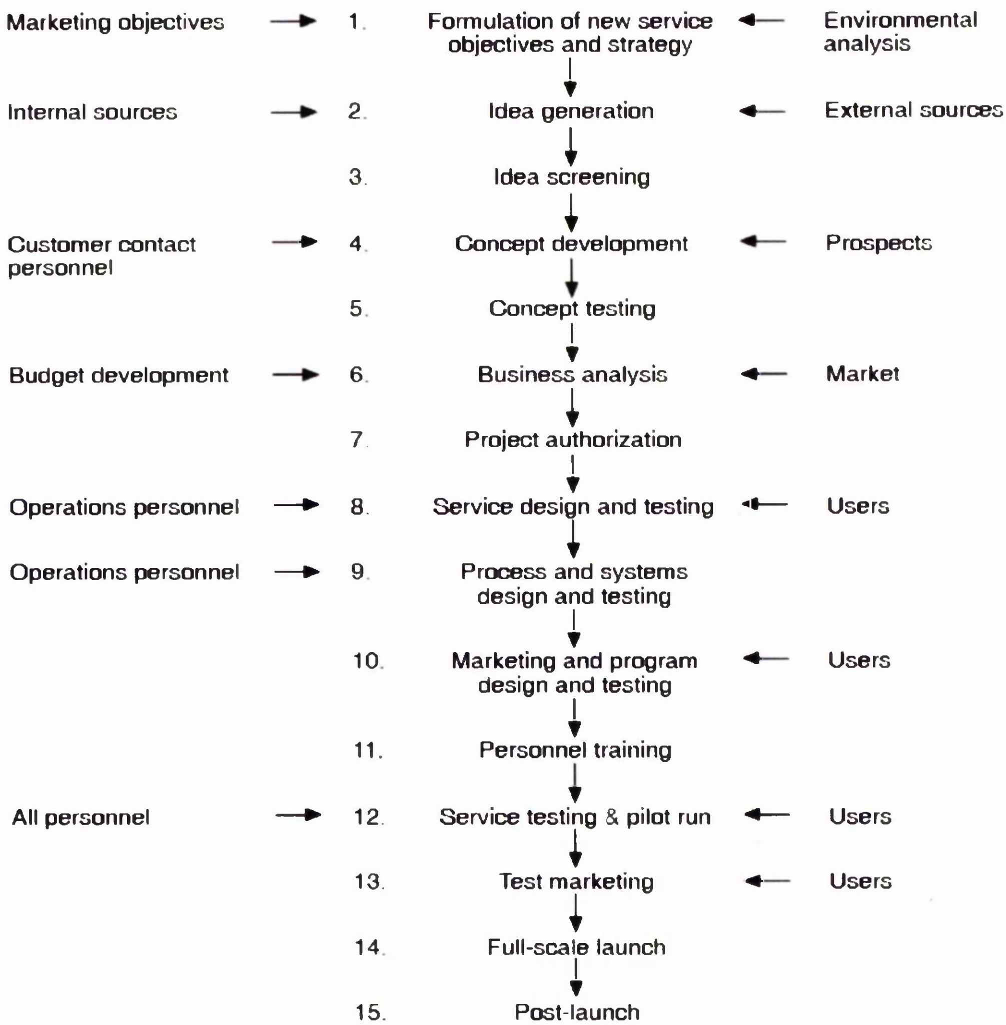


Figure 4.3 New Service Development model (source: Scheuing and Johnson, 1989a)

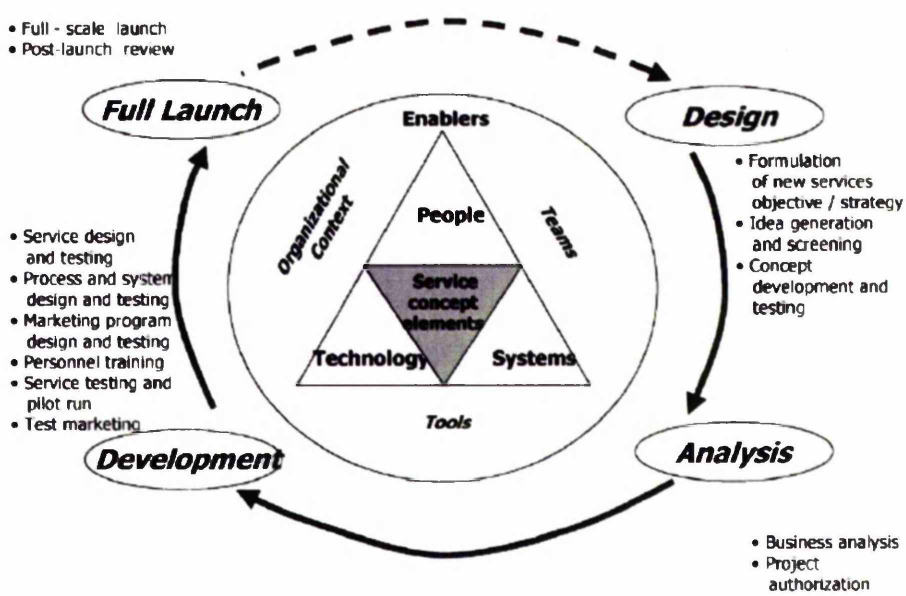


Figure 4.4: The New Service Design Process Cycle (source: Menor *et al.*, 2002)

From the models illustrated, distinct stages in the service design process can be summarized as:

1) Strategic Assessment
2) Concept Development
3) System Design
4) Testing and Implementation.

In the following sections, the principal activities at each stage are explained, and specific attention is given to design engineering support services.

4.2.2 Activities of Service Design Processes

4.2.2.1 Strategic Assessment

Strategic assessment is crucial to the success of a new service design and has a lasting impact on the firm's profitability and growth (Johne and Storey, 1997). At this stage, service providers need to analyze and determine the skills, resources and capabilities required in the development effort. Bitran and Pedrosa (1998) identified four main activities to carry out at this stage; see figure 4.5.

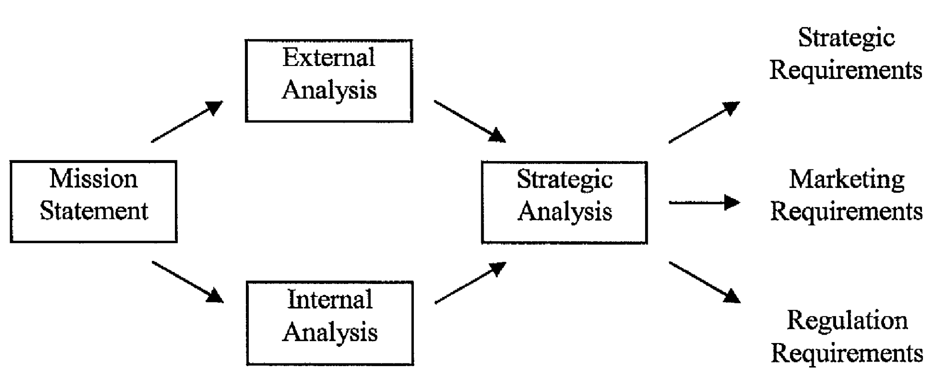


Figure 4.5: Activities of Strategic Assessment (source: Bitran and Pedrosa, 1998)

In fact, the services to be designed are not uniform in terms of degree of innovation. Bitran and Pedrosa (1998) considered that firms may develop new services for reasons ranging from a cosmetic change to serve a known market, to a complete new concept to serve a new market. Johne and Storey (1997), drawing on the categorizations of product development in Booz *et al.* (1967) and Lovelock (1984), suggested six types of new service development; see table 4.1.

Types of New Service Development	Description
New-to-the-world products	New products that not only represent a major challenge to the supplier, but which are also seen to be quite new in the eyes of customers
New product lines	New products which represent major new challenges to the supplier
Additions to existing product lines	New products that supplement a company's established product lines, so rounding-out the product mix
Improvements and revisions to existing product	New products that provide improved performance
Repositionings	Existing products that are targeted to new markets or market segments
Cost reduction	New products that provide similar performance at a lower cost of supply

Table 4.1: Types of New Service Development (adapted from Johne and Storey, 1997)

At the 'Mission Statement' stage, service developers need to set up the target from both the technical and business perspective to determine the new service development type. Each type of project imposes demands on a company's capabilities, logistical system, organizational structure, marketing, and so on, that must be taken into account in the development project. External analysis consists of identifying trends, threats and opportunities in the industry by gathering data about suppliers, customers, competitors, substitutes, regulations and other relevant aspects. Internal analysis is concerned with determining how the new service fits into the firm's current offering and how it will impact the firm's operations. At the end, strategic analysis weights the information generated by the internal and external analysis to generate the outputs listed in table 4.2.

Outputs	Description
Marketing requirements	e.g. business opportunities, market segment and positioning
Strategic requirements	e.g. use of a certain technology, expansion into a new market, etc.
Regulation requirements	e.g. health, safety requirements for the new service, etc.

Table 4.2: Outputs from Strategic Assessment

SSS development in the context of FPs aims to benefit the use of equipment throughout its life cycle. Thus, it belongs to designing 'Additions to Existing Product Lines' and needs to consider the influences from the hardware part, for example the amount of hardware under service policy and the service rate of each piece of equipment. Its strategic assessment is normally stressed when considering the whole package offering.

Based on the information obtained from the strategic assessments, the decision can be made as to whether to proceed with the development of the proposed service. Assuming that the decision to go ahead has been taken, the service design enters into the concept development stage.

4.2.2.2 Concept Development

The concept development phase aims to generate a detailed description of what is to be done for the customer and how this is to be achieved. Edvardsson and Olsson (1996) indicate that the service concept should cover a description of the needs and wishes of the customer and a definition of how these needs and wishes are to be met and fulfilled expressed in terms of the core and supporting services required. In the context of functional products, it needs to clarify the service level that customers want and obligations that the providers need to undertake, and propose a schematic solution.

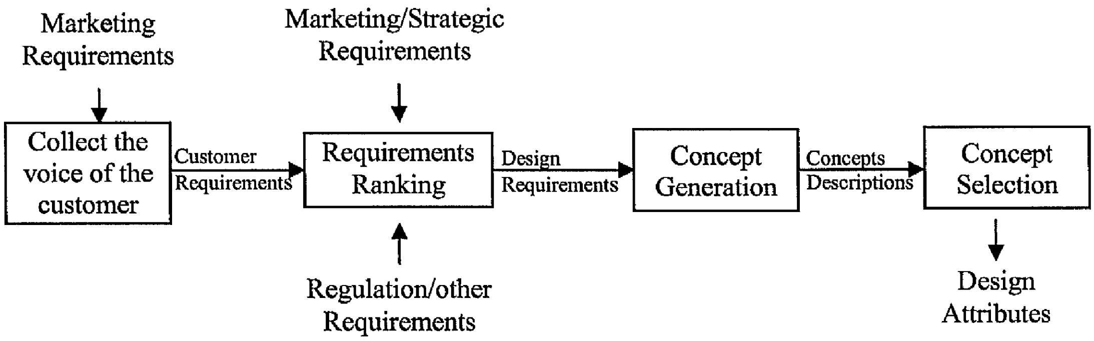


Figure 4.6: Activities of Concept Development (adapted from Bitran and Pedrosa, 1998)

Drawing on the methodology 'Concept Engineering' proposed by Burchill and Shen (1992), Bitran and Pedrosa (1998) pointed out four principal activities of this stage; see figure 4.6. This stage serves as a medium between customer and the service developer and it requires the developer to fully and accurately understand customer needs and wishes. It is critical in avoiding service market failures (Hollins, 2006), that is, the service does not meet customer requirements. The Quality Function Deployment (QFD), proposed by Griffin and Hauser (1993), can facilitate communications between customers and service developers. QFD documents data in a clear, visual manner, enabling ease of use and interpretation; see Griffin and Hauser (1993).

The identified requirements will be refined and ranked. Many ranking schemes have been proposed (Von Hippel, 1986; Burchill and Shen, 1992; Griffin and Hauser 1993; Edvardsson and Olsson, 1996). Two of them are shown in table 4.3. Similarly, in the context of functional products, providers need to weight and prioritize client requirements so as to optimize the usage of resources and capabilities.

Customer Requirement Ranking Schemes	
Griffin and Houser (1993)	Burchill and Shen (1992)
Basic needs: what a customer assumes a supplier will do	Must-be: minimum requirement
Articulated needs: what a customer will tell you that he or she or they want a supplier to do	Attractive
Exciting needs: those needs which, if they are fulfilled, would delight and surprise the customer	One-dimensional: the more of the attribute the better
	Indifferent
	Reverse: the more of the attribute the worse

Table 4.3: Customer Requirements Ranking Schemes

Concept generation is the step where the service developer transfers statements of customer needs that are written in natural language into attributes and functions that the service delivery system will be required to undertake. As Ulrich and Eppinger (1995) note, a particular specification should invariably consist of a metric and a value for the metrics; with the 'value' able to take one of a number of forms including a numeric value, range inequality, etc. During concept generation a sketch of attributes, functions, products and services is proposed.

Ulrich and Eppinger (1995) define concept selection as 'the process of evaluating concepts with respect to customer needs and other criteria, comparing the relative strength and weaknesses of concepts, and selecting one or more concepts for future investigation or development'. The best concepts are given scores based on the assessment of how well they meet their design attributes, weighted by the relative importance of the attributes. The one or two concepts with the best scores proceed to the next stage.

In the SSS design, the linear concept creation procedure (figure 4.6) is normally inadequate to identify precisely what the customer wants, because it involves many considerations including the client's usage of the hardware equipment, client-expected service levels, client-preferred maintenance and repair schedule, and providers' capabilities; trading off between these considerations necessitates an iterative process in which clients are deeply involved.

4.2.2.3 System and Component Design

This phase of service development is concerned with the identification and integration of the required sub-systems and specific components that go to make up the design attributes (Edvardsson and Olsson, 1996). Acknowledging that a system can be divided into sub-systems or components that interact to achieve a certain purpose, Bitran and Pedrosa (1998) state that the 'architectural knowledge' and component knowledge are crucial at this stage.

Harrison and Petty (2002) consider that this process is to some extent abstract as it does not involve the creation of a physical system, selection of hardware or detailed specification of software, but the creation of logical, formal models to represent function and give a definitive representation of a system.

The design task can be split into sub-tasks. Each identified sub-system and component is then successively decomposed into sub-systems. Bitran and Pedrosa (1998) consider it an iterative process concerned with the refining of attributes and specifications. As the decomposition takes place, sub-system specifications are drawn up, viabilities are assessed and designs produced. This process is illustrated in figures 4.7 and 4.8.

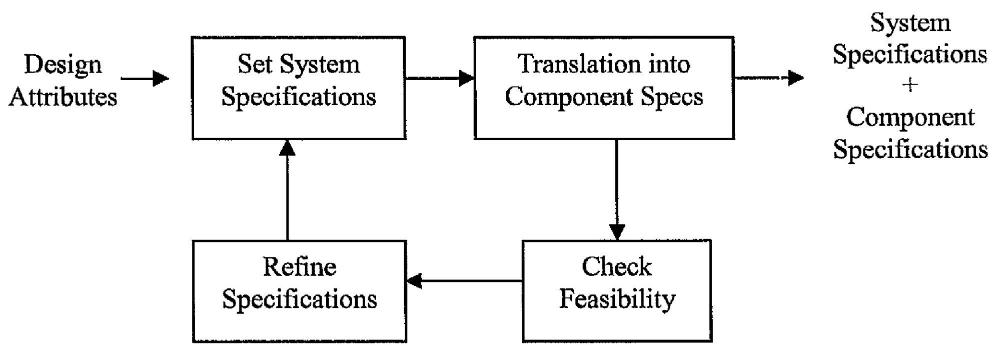


Figure 4.7: Activities of System Design (source: Bitran and Pedrosa, 1998)

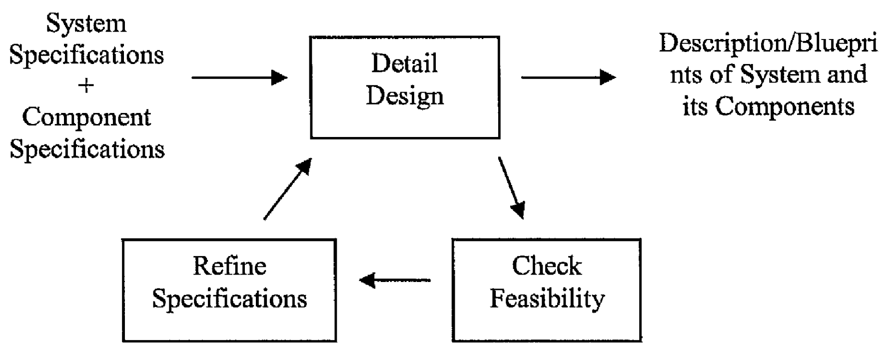


Figure 4.8: Activities of Component Design (source: Bitran and Pedrosa, 1998)

In the context of FPs, each sub-system normally represents a specific function and influences client-preferred service levels; therefore, identification of required sub-systems and components is usually incorporated in the concept creation phase. Component design is concerned with setting up performance specifications for each activity and their refinements are carried out in the testing phase; thus, component design is usually implemented at the testing stage. But analogously, both are iterative procedures.

4.2.2.4 Concept Testing and Implementation

Shostack (1984a) states that the outcome of the service development process is the set of clearly defined activities needed to generate the service. These activities are described by a set of logical, formal models (Harrison and Petty, 2002) that enable the proposed service to go into operation; however, before the newly proposed service is launched, testing is necessary. Testing enables the supplier to investigate the performance of the new service before committing to full-scale implementation. With results of the trials, the supplier can assess whether the service meets the original objectives in terms of expected and required performance and function, whilst at the same time highlighting those areas where further re-design or refinement are necessary or would be of benefit.

The testing phase for a service is different from that of a conventional product, which often produces prototypes to evaluate hardware performance.

Producing a prototype for a service will consume almost equivalent resources to a full-scale release. The alternative is implementing pilot schemes for services. But is has been recognized by Wheelwright and Clark (1992) that, for the testing of an intangible concept such as a service to be of significant benefit, it must be undertaken under market conditions or, if this is not possible, then as close to actual market conditions as is obtainable. This enables customers to understand the service concept and provides the supplier with an invaluable opportunity to test the service in the market place and obtain feed-back from customers. Murphy and Robinson (1981) have come up with a set of criteria testing services from the marketing perspective to evaluate whether a prospective user:

(a) Understands the idea of the proposed service,
(b) Reacts favourably to it and,
(c) Feels it offers benefits that answer unmet needs

Although it has been widely recognized that service testing is significant and beneficial, testing of service systems is seldom carried out. One important explanation is that because service innovation can not be patented, the suppliers of new services worry that rivals could make use of information obtained from the tests and even copy or implement a similar service utilising the test information. Another explanation is that the cost of service failure may be relatively low compared to the testing costs.

Service implementation is split by Shostack (1984b) into three phases:

Phases	Main Tasks
Operations plan implementation	Put operation functions in place and test; correct any problems; Draw up a detailed plan of implementation including dates and schedules, culminating in the required launch date of the service;
Communication strategy implementation	Advertising, Publicity, Promotion
Introduction to the Market	Collect and Monitor feedback

Table 4.4: Phases and Main tasks in New Service Implementation

Training is also highlighted during implementation. As argued by Scheuing and Johnson (1989), training should not be confined to service personnel; customers also need to be taught how to use service innovations.

In the context of FPs, the existence of an objectively agreed system performance level gives rise to the possibility of using computer-based virtual testing, for example Monte Carlo simulations, to test system performance through evaluating performance indicators such as system reliability.

4.3 Design Services of FPs

4.3.1 The Processes of Designing Engineering Support Services

This section concentrates on the development of SSSs for FPs. In FP design, the linear process is not adequate to address client requirements. In comparison with designing general services, concept creation for service systems of FPs needs to be a continuously interactive process between the client and provider in that the available options are not readily apparent; therefore, the client should be involved in the concept design process. The concept design process is complete when the client and provider are satisfied with functionality and cost, such that the concept is deemed feasible, and a decision to a contractual agreement is made.

Another difference is that service design in the context of FPs needs to take into account effects of the core hardware, for example the hardware's failure rate. In concept design for services in general, the focus is entirely on the service and the design of the hardware is not considered. This is different for FPs as hardware is an integrated part of the product and therefore the capabilities of the hardware are considered as part of the concept creation process (Alonso-Rasgado *et al.*, 2004).

Alonso-Rasgado *et al.* (2004) identify five stages in the design of service support systems for functional products, as shown in figure 4.9.

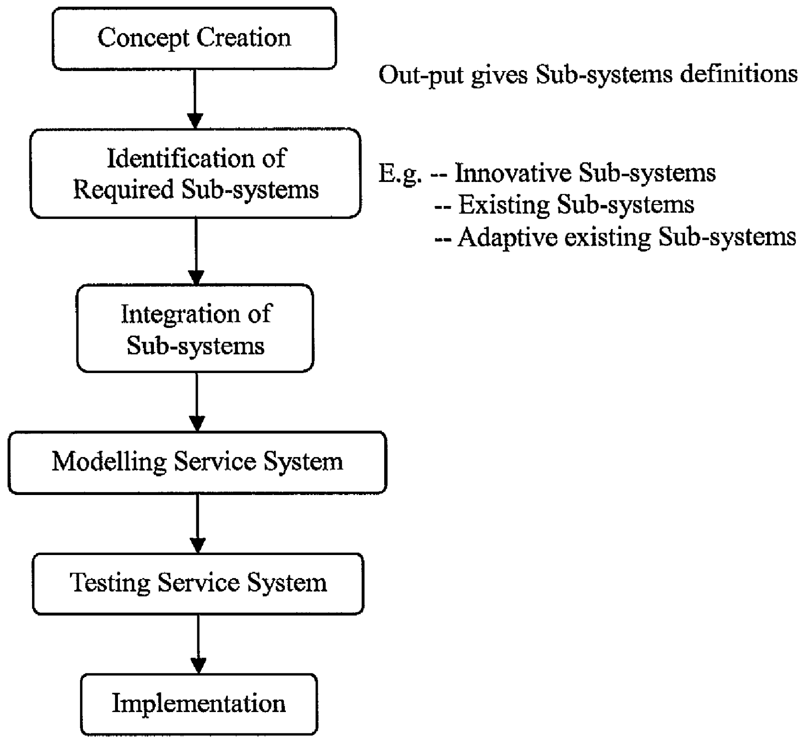


Figure 4.9: Five Stages in Designing Services for FPs (adapted from Alonso-Rasgado *et al.*, 2004)

Although similar to those proposed in designing general services, the model proposed by Alonso-Rasgado *et al.* puts emphasis on two aspects: integrated concept creation and service system modelling. These two concerns are explained below.

4.3.2 Integrated Concept Creation

In FP design, an integrated concept design process is required in which customer aspirations, product potential and concept development should be explored simultaneously. During interactive concept creation, customers are encouraged to take part in order to explore the possibilities together with the supplier. As shown in figure 4.10, the process begins as the clients propose a set of requirements. To respond, service providers will develop a functional model and generate a proposal which acts as feedback to clients. After evaluating the potential solutions, the client will further clarify requirements, which will be responded to with appropriate solutions. This process should only end when both parties agree with the functionality and cost, such that the concept is deemed feasible, and a decision to make a contractual agreement is made.

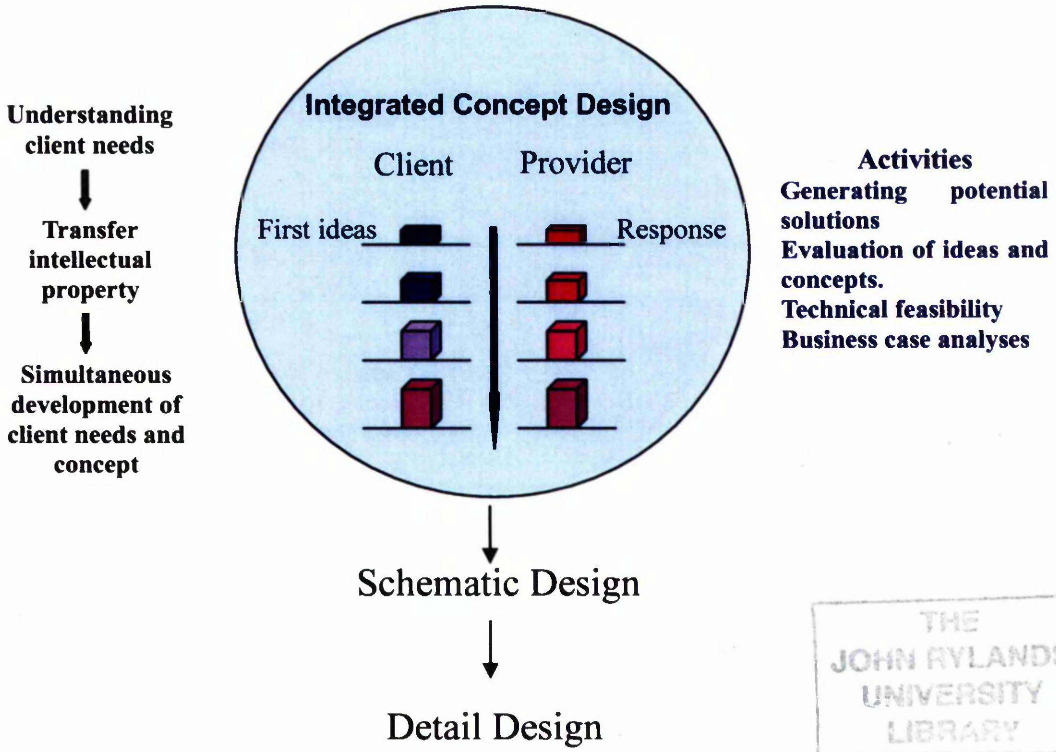


Figure 4.10: Integrated Concept Design

The identification of the required service sub-system process is carried out simultaneously with the service concept creation phase. This is slightly different from that found in the development of general services, where the sub-system required is identified at the system design stage. Alonso-Rasgado *et al.* (2004) consider that in the concept creation phase, service providers are able to tell if an innovative sub-system is required or an existing sub-system can be employed to perform the service system. In some cases, existing service sub-systems can be brought together to form a new service sub-system.

4.3.3 Service System Modelling

Models play a significant role throughout the service design process. Bitran and Pedrosa (1998) recognized the pivotal role that modelling plays in the design quality of a product or service, and commented that being able to approximate the relevant behaviours of the system under development enables designers to assess the feasibility of the proposed service solution. They attempt to demonstrate how it can be of benefit in the design of intangible elements by presenting modelling applications that originate from operational research, and indicate that such modelling can be applied in two contexts in service development: performance evaluation and optimization. Examples include Ittig (1994), who investigates the impact of having demand as a function of average waiting times in the design of the number of check-out counters at a supermarket; Sze (1984), who describes a queuing model that predicts average delays in a telephone operator system; and Cherng *et al.* (2005), who describe a dynamic model for determining the optimal number of tollbooths.

In fact, modelling can assist service development better than evaluating trade-offs between numeric variables. It is necessary to visualize SSS's many intangible elements through appropriate modelling techniques. Clark *et al.* (2000) recognize that it is important that all parties involved in service development, both customers and suppliers, have the same vision or iconic mental representation of the proposed service, to ensure that the service concept is understood and interpreted in a similar manner by all. If this can be achieved then the gap between what the customer expects from the service and what the service actually delivers in practice can be minimized. As stated by Shostack (1981), the first step towards rational service design is a system for visualizing this phenomenon. To this end Shostack (1981, 1984a, 1984b) proposed two approaches: Molecular Modelling and Service Blueprinting.

However, in practice, apart from the applications of the queuing model, little research effort has been undertaken into this service modelling (Johne and Storey, 1996; Bitran and Pedrosa, 1998; Alonso-Rasgado *et al*, 2004b). Reasons for this include:

1) Fundamental differences between new product development and new service development
2) The lack of an appropriate methodology for services

3) The volume of detail and complexity of the proposed service
4) Time and resources required

A major difficulty is that with service modelling, human behaviour is an important factor that must be taken into account (Johne and Storey, 1996; Bitran and Pedrosa, 1998). However, inability to accurately describe human behaviour, which does not follow defined laws like the physical behaviour of hardware systems, causes one of the main obstacles in the modelling of service systems.

For FPs, service system modelling can be divided into two groups: static modelling and computational modelling. Static modelling has its equivalent in the design of general services. Static system modelling is indispensable in that it forms the basis for building computational models and simulations. Alonso-Rasgado *et al.* (2004) suggested the use of computational simulation after the modelling of the service system. Using computational simulation can act as a tool for studying the performance of the proposed system. Degraded performance sub-systems can be identified and can be modified accordingly. These modifications can be checked by re-running simulations. Using the idea of computational simulation, the service system can be tested with different resource structures so as to give a general idea of how the service performs under certain conditions. This can be done prior to the pilot scheme or full-scale implementation.

4.4 Service System Modelling Methodology

Static system modelling aims to generate a definitive representation of a system. Effective static modelling is critical for the precise testing of service systems with simulations. This section reviews main approaches to the modelling of service support systems and discusses their advantages and disadvantages in the context of functional products. The approaches include:

- Flowcharts
- Molecular Modelling

- Service Blueprinting
- Structured Design and Analysis Technique (SADT)
- System Diagrams.

4.4.1 Flowcharts

A flowchart is a schematic representation of an algorithm or a process. It is commonly used in business, programming and economic presentations to help the audience visualize the content better, or to find flaws in the process. Flowcharts are probably the simplest symbolic technique to model the operations of a service by connecting involved elements such as input, process and output; see figure 4.11.

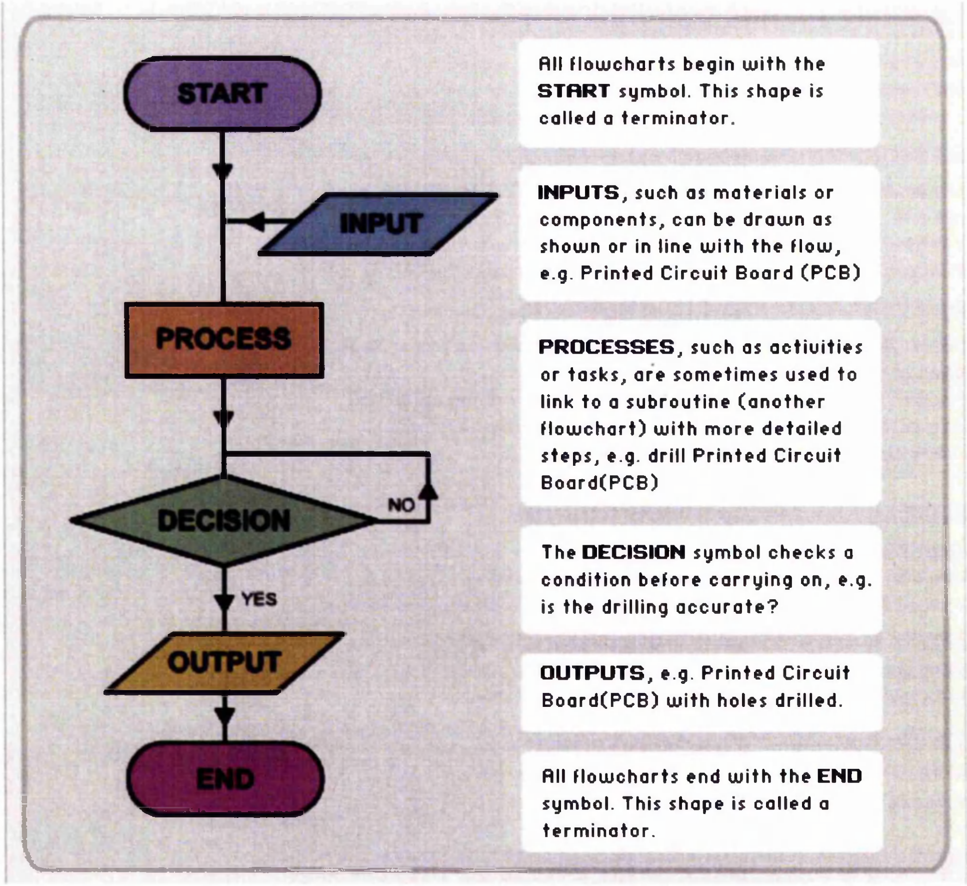


Figure 4.11: System Flowchart Symbols (source: www.bbc.co.uk)

The main advantage of using flowcharts is that they are simple and require little experience for clients to obtain a good understanding. In addition, for service designers, the operations of flowcharts are intuitive and easy to explain.

The major disadvantage of flowcharts is that they can not cope with realistic, complex situations. More fundamentally, the flowcharting technique does not correspond to real-word systems that display structure at a number of levels. Finally, flowcharts are not easily translated into information systems. Despite their limitations, flowcharts can still be a useful and quick way of describing systems (Harrison and Petty, 2002).

4.4.2 Molecular Modelling

The molecular modelling approach proposed by Shostack (1981) depicts a comprehensive picture of both tangible and intangible elements in a service. The approach is flexible and utilizes symbols to denote views product/service combinations as atoms or entities connected in a unique molecular configuration.

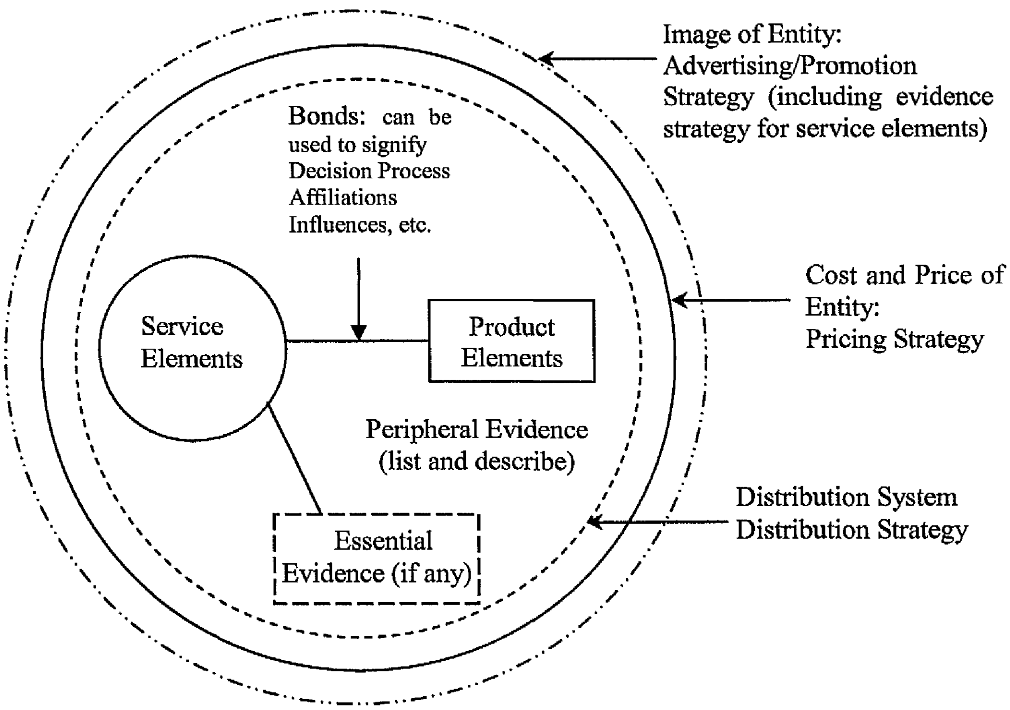


Figure 4.12: Components of a Completed Molecular Model (adapted from Shostack, 1981)

Figure 4.12 illustrates the structure of a completed molecular model where the product and service elements have been identified, the bonds between the elements have been described and the tangible and intangible items distinguished. The remaining elements, namely the entity's distribution system, the entity's cost and proper price, and the advertising and promotion of the entity, are shown ringing the entity and reflecting their relationship to the entity and the order in which they should be dealt with.

A simple application, as shown in figure 4.13, illustrates a corner shoeshine concern to which the principles of the molecular modelling approach have been applied. The following attributes apply in this case (table 4.5):

Elements	Description
The service element:	Shoeshine
Product element:	Wax
Distribution:	Corner of the sidewalk
Image:	Convenient, fast and inexpensive
Price:	50 pence

Table 4.5: Specifications of a Corner Shoeshine Shop

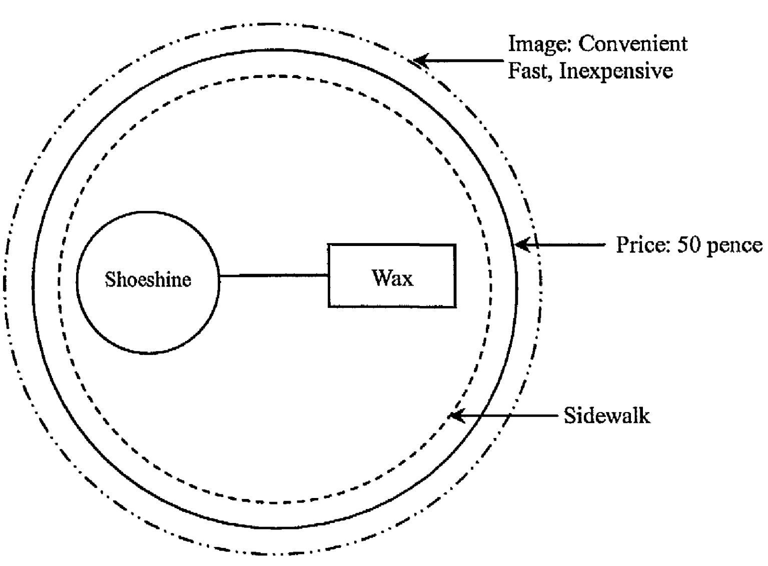


Figure 4.13: A Molecular Model for a Shoeshine Shop (adapted from Shostack, 1981)

The approach can be utilized to obtain a comparative scale of dominance which arranges entities according to their overall makeup. The molecular modelling approach has the following advantages:

1. Allows full consideration of the service elements and the product elements.
2. Provides a structure for identifying and visualizing all parts of any complex market entity.
3. Gives an indication of how the system will behave or react when any change or rearrangement occurs in the system.

The main disadvantage of this approach is that it does not illustrate the internal activities in a dimension of time, which makes it unable to serve as the preparation of service computational modelling.

4.4.3 Service Blueprints

Acknowledging that services are fundamentally processes, Shostack (1984a) developed the service blueprint approach which attempts to map the processes of a service in an objective and explicit manner while also capturing all the essential functions to which marketing applies. The service blueprint is a type of PERT Chart which is usually used for project scheduling as opposed to process description. The service blueprint shows time dimensions and costs for each process of the service and shows the minimum time needed to complete a service. Shostack (1984a) identifies the basic requirements of a service blueprint as being:

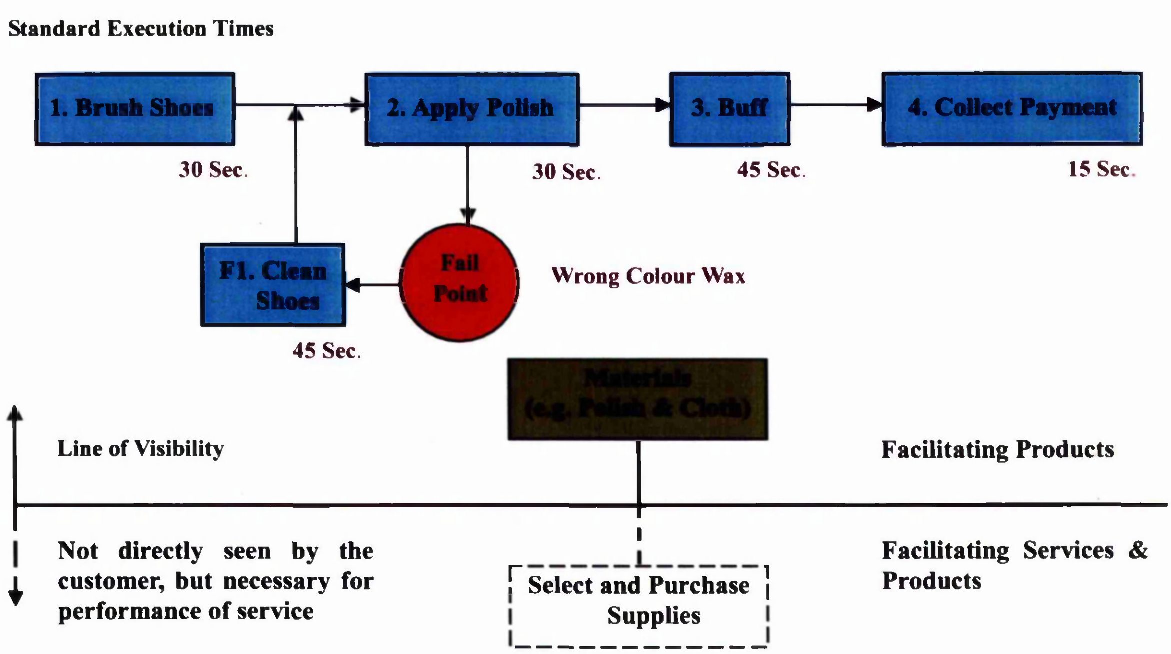
1. It must show time dimensions in diagrammatic form, as does the Project Evaluation and Review Technique (PERT charting; for details see http://en.wikipedia.org/wiki/Program_Evaluation_and_Review_Technique);

2. It must identify all main functions and sub-functions of the service;

3. It must define the tolerance of the model, i.e. the permissible variation from the blueprint's standards that can be allowed in executions without

affecting the consumer's perception of overall quality and timeliness.

The actual process of designing a blueprint involves the consideration of several issues (Shostack, 1984a) including identifying the processes that constitute the service, isolating the points in the system where failure may occur, establishing time frames including standard execution times and deviations, and analysing profitability which involves establishing a time-of-service execution standard that precludes unprofitable business and maintains productivity. If these functions are performed by people, a work chart should be constructed. All inputs and outputs of functions must be shown. Figure 4.14 shows a blueprint for the relatively simple example of a shoeshine service.



Standard Execution Time: 2 Minutes
Deviation Tolerance: 3 Minutes (Inter-cycle tolerance=1 minute, Extra-cycle tolerance=2 minute)
Total Acceptable Execution: 5 Minutes

Figure 4.14: Blueprint for shoeshine shop (adapted from Shostack, 1984a)

The 'line of visibility' shown separates those parts of the service that are visible to the customer from those which are not. Execution times and tolerances are shown. The information displayed in the blueprint can aid the supplier in setting service standards and tolerances. The total tolerance is divided into two categories:

1) Inter-cycle deviation tolerance occurs within the service process itself, if any of the steps take longer to perform than the time specified in the blueprint.
2) Extra-cycle deviation tolerance occurs outside the service process, e.g. waiting two minutes in the queue to be served.

Both types of deviation tolerance affect customers' perception but usually inter-cycle deviation affects profitability.

The advantage of service blueprints lies in that it allows a company to test its assumptions on paper and thoroughly work out the bugs. Shostack (1984a) identified the potential benefits of service blueprinting as:

- It provides a visual and quantitative description of any service element;
- It allows a service to be created on paper;
- It allows the marketer to know exactly what is being tested, e.g. deviation tolerances, fail points, consumer values associated with specific functions, etc. (this in turn facilitates the following point);
- It can be used to mock up a prototype or pilot service that can provide the marketer with concrete actionable feedback that can be used to modify the service.

Criticisms on blueprints mainly focus on that it rarely takes into account the need for flexibility when things go wrong or occur out of the ordinary. Service blueprints must be prepared to cope with the unexpected (Hollins, 2006). In addition, Johne and Storey (1996) point out that the actual service will deviate from the blueprint in terms of duration, quality and customer satisfaction. This is because service development does not result in the production of the service itself; rather it produces what has been termed the service 'pre-requisites' (Edvardsson and Olsson, 1996). The actual service offered is only produced when the customer interacts with these pre-requisites.

Lovelock (1984) adds a note of warning about service blueprints, stating that in many cases they can fail due to the fact that operational efficiency can dominate over the concerns of the customer. The solution recommended by Lovelock is to produce two sets of blueprints: one from the company's perspective and a second one from the customer's point of view.

From the perspective of preparing for service computational modelling, the main drawback of blueprints lies in that blueprints can not display the service system hierarchically, which usually leads to a cumbersome chart when coping with complex services comprising a large number of activities. This leads to the difficulty of analyzing the activity's inputs and outputs and their logical relationships.

4.4.4 Structured Analysis and Design Technique

Structured Analysis and Design Technique (SADT), also known as $IDEF_0$ (Integration Definition for Function Modeling), is graphic languages describes activities as the blueprinting. The methodology was developed during the 1960s and early 1970s to describe complex systems and is well known for designing a training system for the United States Air Force. Since its introduction into the market in 1973 it has been utilized in diverse industries such as aerospace, telecommunications, and software development (Guinet, 1990; Marca, 1991; Siltala, 2004).

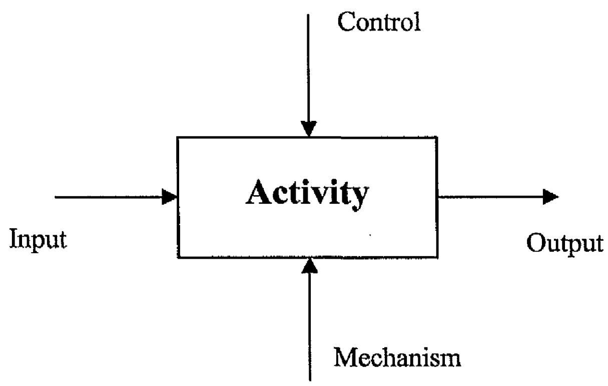


Figure 4.15: Schematic representation of the SADT structure activity box (adapted from Congram and Elpeman, 1995)

The general structure of SADT is illustrated in Figure 4.15, where the sequential process is that under control and input is transformed into output by the mechanism; see table 4.6.

Nomenclature	Explanation
Activity box	A verb phrase that describes the activity
Input	Required information for the activity to implement
Output	Information or tangible evidence generated by the activity
Control	The trigger for an activity; often refers to information that directs what activities do, constraints that limit activities
Mechanism	Represents the physical elements (e.g. employees, customers, machinery) through which things get done,

Table 4.6: Basic nomenclatures describing an activity in SADT (Congram and Elpeman, 1995)

Congram and Epelman (1995) consider that existing service modelling approaches tend to be flow-diagram in their approach and do not form a sufficiently disciplined process for describing services. They state that SADT is especially effective for services for reasons including:

- SADT focuses on activities, the building blocks of services. SADT models can help employees at every level to understand what happens in delivering a service.
- SADT is a methodology that structurally provides important attributes of service description as: who or what performs the activity ('mechanisms' in SADT terms); and what guides or limits the activity ('controls').
- SADT is valuable in improving internal communication because the model-building process includes a protocol to involve employees, other people who perform activities, management, and customers.

In terms of services, SADT has to date been primarily used to model back-office operations. The SADT modelling process results in a set of interrelated diagrams that collectively describe a system. The top diagram summarizes the diagrams below, which are arranged hierarchically and become increasingly more detailed at every successive level.

In order to illustrate the application of SADT, Congram and Elpeman (1995) used individual tax return processing as an example. The description starts with a top level diagram (figure 4.16) that summarizes the general activity, and figure 4.17 illustrates the decomposition of the top level of 'process individual tax return' (see figure 4.17) into its four components: *Manage processing*, *Prepare draft*, *Review draft*, *Copy and sign final return.* Each of the four activities comprised in the activity 'process individual tax return' may be decomposed into a more detailed set of activities.

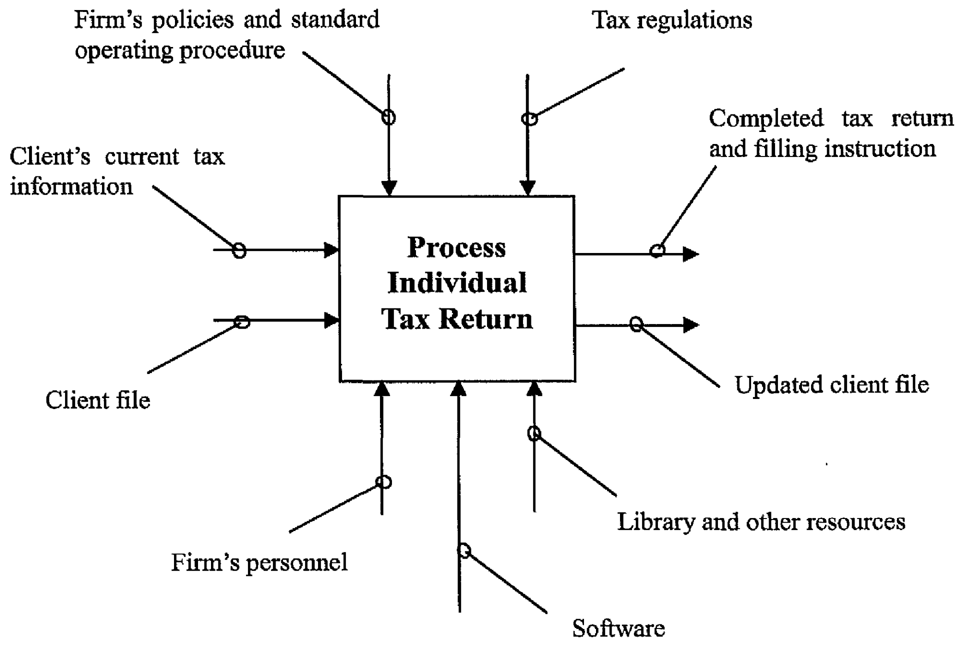


Figure 4.16: Top Level of Individual Tax Return (adapted from Congram and Elpeman, 1995)

The authors identify the limitations of SADT in modelling services, that it is a methodology within a static modelling paradigm that can deal with the representation of the structure of a system but can not deal with the system behaviour over time. They point out that SADT is not a simulation tool but nevertheless it can help in the preparation for simulation. They conclude that it is a useful method that brings discipline to service system modelling, helping to provide a methodology for reaching organizational consensus about service processes.

Harrison and Petty (2002) consider the principal disadvantage of SADT is that preparation of the diagrams can be time consuming; the main advantage is the precision and clarity of the final result.

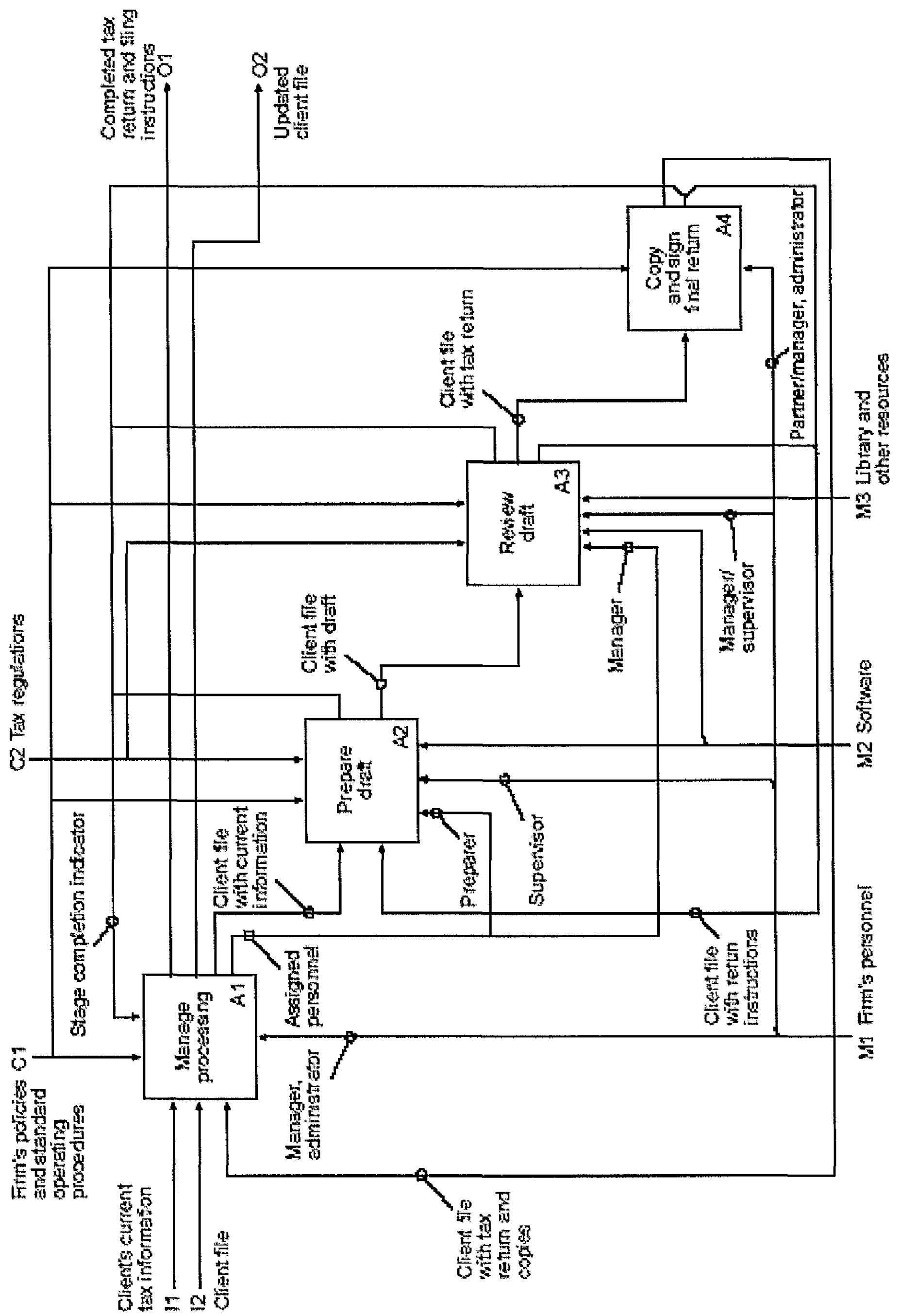


Figure 4.17: Decomposition of activity 'process individual tax return' (source: Congram and Elpeman,1995)

4.4.5 System Diagram

The system diagram, also known as the function block diagram, is a graphical representation for a system from the functional perspective (Harrison and Petty, 2002). The most basic form of a system diagram is depicted in figure 4.18. The basic unit consists of an input, an activity (sometimes also known as a function) and an output. The input and output are connected to the function by using connection lines. In some cases, multiple inputs and outputs of a function are necessary. Typical units of system diagrams are similar to typical activities described in section 3.3.

Figure 4.18: Basic Unit of a System Diagram

The development of the system diagram involves the identification of the functions in the service system formulated during the service concept creation phase. By identifying and grouping similar functions in the functions table, and matching the inputs and outputs from the identified functions, the system diagram can be constructed. Grouping of similar functions creates sub-systems and leads to formation of a hierarchical view of the system diagram; see figure 4.19.

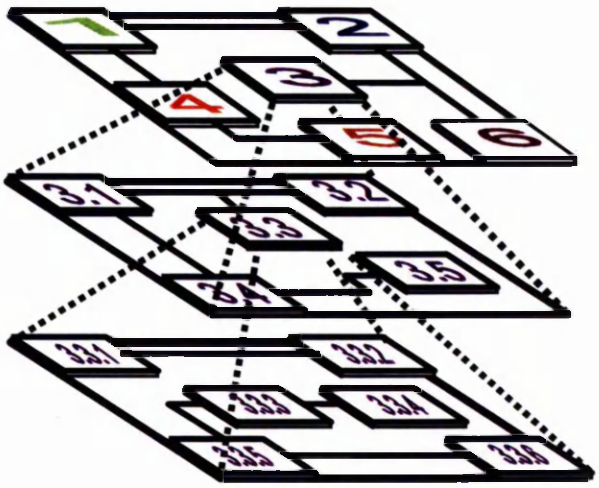


Figure 4.19: System Diagram Hierarchy (adapted from Harrison and Petty, 2002)

The system diagram is an ideal tool for building the static model of service for reasons including:

- it is activity-oriented;
- it can hierarchically display the service systems;
- it avoids differentiating 'control' and 'mechanism' from inputs;
- there are no special limiting rules (no more than eight activities lie in one level);

4.5 Challenges in the Design of Engineering Support Services

4.5.1 How to Achieve Customization of Services

In today's intensely competitive commercial world, it is wise to consider customers' requirements during the service design process, which to some extent guarantees profits. Hill *et al.* (2002) categorize services into three groups in terms of their degree of customization: Customer-routed, Co-routed and Provider-routed. In their words, customer-routed processes offer the customer relatively broad freedom to select from many possible options, while provider-routed processes constrain customers to follow a very small number of possible routes. Co-routed processes offer customers a moderate number of routes.

As previously mentioned, co-routed services can bring added value to both service providers and customers. This gives rise to a requirement for an innovative way of designing services. Customer requirements must be fully understood so that the service designer can talk to customers during design processes, which are usually complex. Hill *et al.* (2002) consider that using some IT technology can help achieve high customization by giving customers more chances to design and simulate their desired services. E-commerce was given as an example.

In the provision of FPs, suppliers tend to provide flexible solutions from which clients can select according to different needs. For example, clients can choose comprehensive support services or specific functions which complement their own capabilities. The key is to provide a platform for both suppliers and clients to work jointly to achieve an agreed solution, but verifying the solution's feasibility is also necessary. Customization could be reflected in providing a customer-preferred support level, customer-preferred system performance and customized service schedule. These have to be allowed for in the design process.

4.5.2 Human Issues in Service Design

In service systems for FPs, as in general services, human behaviour provides the biggest source of uncertainty. This has been recognized by many authors when discussing service modelling techniques. Human behaviour is deemed to be an important reason for the lack of service modelling tools.

As mentioned in chapter 2, evaluating the reliability of a service system needs to model the performance uncertainties incurred because of human factors. Therefore, during the service development, the uncertainties have to be taken into account.

4.6 Summary

From a thorough review of the literature available on service design, it can be observed that the service design process is similar to the hardware design process. Designing supporting services for FPs shows several deviations from that for general services. The strategic assessment is integrated with the design of whole-package provision. Compared with the linear sequence of activities implemented in general service concept creation, the concept development of engineering support services demands the deep involvement of clients, and an iterative interactive process is beneficial. The system and component design phases are not independent as in general service design, but are incorporated into the concept development and testing phases. Service testing is beneficial in that it enables service developers to evaluate system performance and modify possible drawbacks before the design is fully launched. Compared with the pilot scheme usually adopted in testing general service design, engineering support services may have to be tested in a virtual environment because of the existence of relatively objective system performance specifications. Established system modelling methodologies can be utilised in preparation for building computational models of service systems.

The main challenges, customization and uncertainties of human performance, lie in the design of service support systems in the context of FPs and call for an innovative approach to address them.

Chapter Five

'Bottom Up' Service Design

5.1 Introduction

Designing services normally follows a similar process to that of hardware design; that is, a process that starts from customer needs, through concept and detail design to system testing. Designing SSSs for FPs may require providers to create a new system or to adapt an existing one, depending on the level of innovation required and client business needs. Five stages are identified in the design of such systems by Alonso-Rasgado *et al.* (2004): Concept creation, Identification of required sub-systems, Integration of sub-systems, Modelling service system and System testing and implementation.

In designing SSSs, identification of required sub-systems is a part of concept design. The procedure could follow a 'Bottom Up' process during which the SSS gradually evolves from individual activities to a set of well specified system diagrams by first identifying each individual element. During the process, it is possible to tailor flexible solutions to meet the diverse client needs.

In this chapter, the main tasks of the 'bottom up' method of designing SSSs are first illustrated, and a model describing their sequential and iterative relationships is proposed. This is followed by the introduction of a computer-aided design (CAD) tool, the System Design Interface (SDI) that aims to facilitate the design procedure. The chapter ends with summarizing the advantages and disadvantages of the 'bottom up' design process.

5.2 'Bottom Up' System Design

5.2.1 Design Process

The 'Bottom Up' design process is shown in the flowchart in figure 5.1.

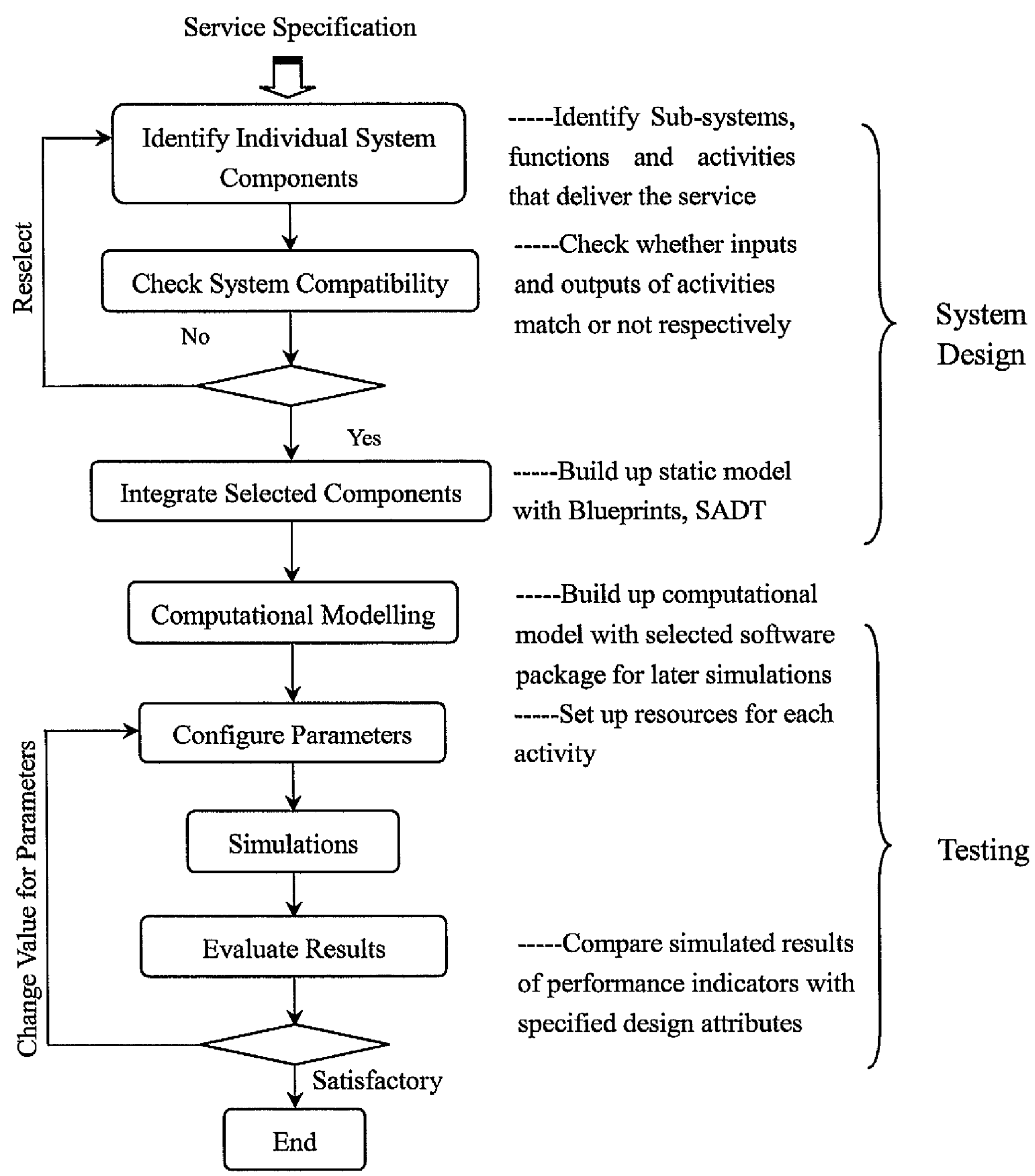


Figure 5.1: 'Bottom Up' Service Design Processes

In a 'Bottom Up' approach the individual base elements of the system are first specified in detail. These elements are then linked together to form larger sub-systems, which in turn are linked, sometimes through many levels, until the system eventually grows to the required level of complexity and completeness.

5.2.2 Identify System Components

The first step using the 'Bottom Up' approach is to identify the elements that are needed. Every individual activity should be defined in detail in terms of, for example, input(s), output(s) and the function to create inputs from outputs. As activities may be undertaken by several companies, such identification is an exploratory procedure and is usually rather time-consuming, especially when developing a SSS with a high level of innovation. In such cases, it is of less help to refer to existing systems.

The identification may be implemented by the FP provider and client jointly as the successful provision of FP requires the full collaboration of a number of parties including FP provider, client and subcontractors. In addition, it offers the client maximum freedom of design, which in turn can help the provider meet the diverse business needs of clients.

It is important to ensure that all parties involved in the service development have the same vision of these functional components (individual activities). As functional components are intangible, it is beneficial to visualize the design process. If this can be achieved then the gap between what the customer expects from the service and what the service actually delivers in practice can be minimized. To this end, a graphic user interface, System Design Interface (SDI), has been developed to facilitate the process. It will be introduced in section 5.3.

5.2.3 Check System Compatibility

After the components have been identified by designers and clients, there arises a pivotal question: whether they are able to work consistently to deliver the defined service; that is, whether all elements of the proposal have compatible inputs and outputs. A compatibility check is therefore the second step in the 'Bottom Up' design process and its purpose is to ensure all identified components have compatible inputs and outputs. Incompatible proposals are required to be reconfigured until a compatibility-verified group of sub-functions and activities are identified. Thus, compatible SSSs are defined as:

> *Systems comprise components that have adequate input(s) to trigger their implementation.*

To verify whether a SSS is internally compatible or not, the principle is to check whether all the inputs and outputs match. Figure 5.2 shows three sample activities with their input(s) and output(s) labelled, which can be combined as the consistent system shown in figure 5.3. Outputs A_{Opt02}, B_{Opt01} and C_{Opt01} become the outputs S_{Opt01}, S_{Opt02} and S_{Opt03} that are external to the system. If activity B is not identified, the system will not be compatible.

The simple paradigm shown in the figure is only for illustrating the principle of compatibility checking and the cases described may not exactly match every real situation. To actualize the specified service, more activities normally need to be identified. Thus, the relationship between their inputs and outputs are not straightforward, which makes it difficult to carry out the system compatibility check on paper. Computer programming is required. Such programming has been integrated with the SDI. For the programming source code, see appendix A.

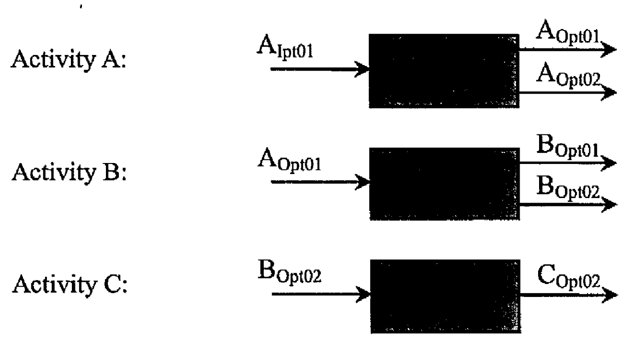


Figure 5.2: Three Sample Activities

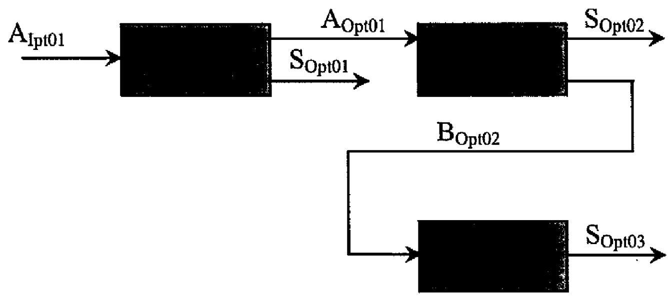


Figure 5.3: The Compatible System

5.2.4 Integrate and Create System Model

The aim here is to create a logical, formal model to represent a system in which the input(s) and output(s) of selected activities or components connect accurately. The process is often referred to as static system modelling.

Several static modelling tools could be employed; see section 4.4. The System Diagram is adopted in this research. Figure 5.4 shows a sample system diagram developed using a powerful software package, 'iGrafx Flowcharter'. Generation of such system charts is for clarifying relationships between identified components and providing a reference for building the computational model.

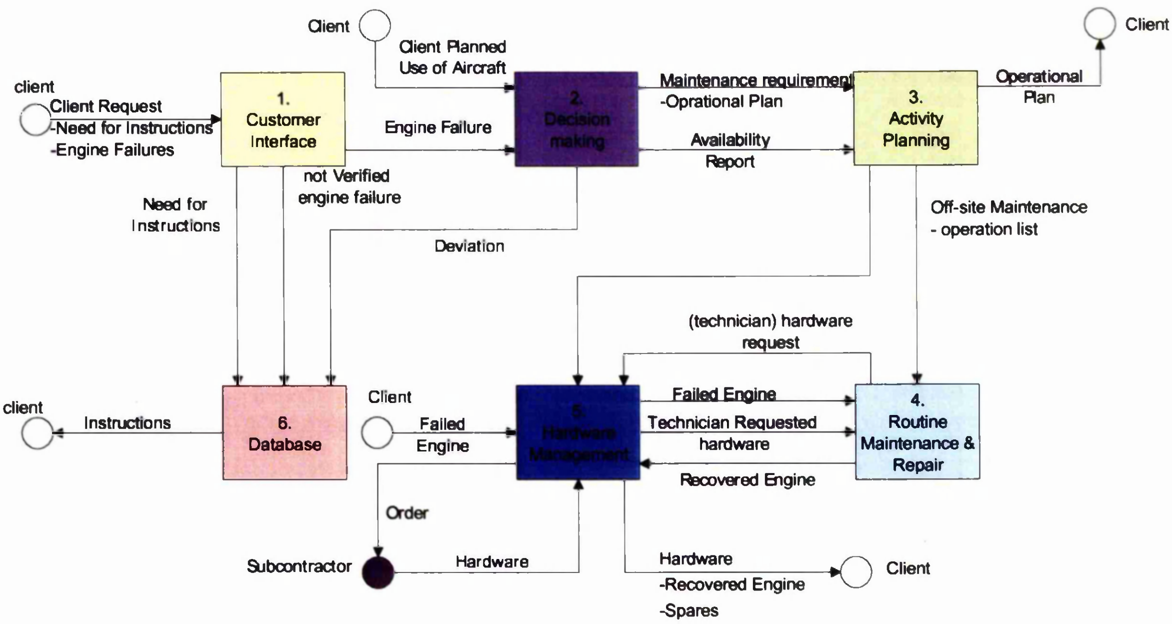


Figure 5.4: Illustration of a System Diagram

5.2.5 System Functionality Test

The system chart generated in the 'bottom up' process may deviate from the theoretical or existing systems, so business risks from the innovation are usually high (Alonso-Rasgado *et al.*, 2006). This necessitates a functionality test of the candidate solution. In such tests, according to Alonso-Rasgado *et al.* (2006), a risk assessment may be carried out to uncover potential threats to an otherwise reliable system.

Simulation-based system functionality testing is beneficial because, compared with traditionally implemented pilot schemes, it provides better protection of the intellectual property from copying by rival companies. Moreover, it costs less time and resources, often an obstacle handicapping service providers in carrying out adequate testing the system 'goes live'. With the aid of computers, modelling and simulating a service system provides a low-cost and effective solution. This stage contains four tasks:

1. System Computational modelling
2. Configuration of Parameters
3. Simulations
4. Evaluating Results.

The computational model was built upon the static service charts that are generated after integrating identified components. It enables simulations to be carried out of the behaviour of the candidate solution over a specified time of interest. The computer software package used in this research is MATLAB®/Stateflow, a widely employed engineering software package. It was chosen for its flexibility in building simulation models and the extensibility of its functionality through the wide range of toolboxes available. The details of building the computational model will be given in chapter 7.

During system testing, the developer can configure parameters of each system component, for example time and resources, for simulations and evaluate testing results. If any component performs poorly, it will be identified and correction can be made; testing of the corrected system can be performed by re-running the simulation models. Such a test will end when the system achieves defined service attributes with a configuration of parameters of each component.

5.3 System Design Interface

A graphic user interface, the System Design Interface (SDI), has been developed using Visual Basic to facilitate the element identification and compatibility check.

The components are identified starting from the root level (lowest level). SDI provides a canvas, the component identification panel, to display the icons of activities; see figure 5.5. By using the button 'Add', activities could be further added with their captions, and input(s) and output(s) designated within the 'Activity Control Panel'. The button 'Delete' erases the unwanted activities while 'Clear' deletes all components shown on the canvas. By clicking the interested component, for example activity 'A', its properties including caption, input and output, etc. are displayed.

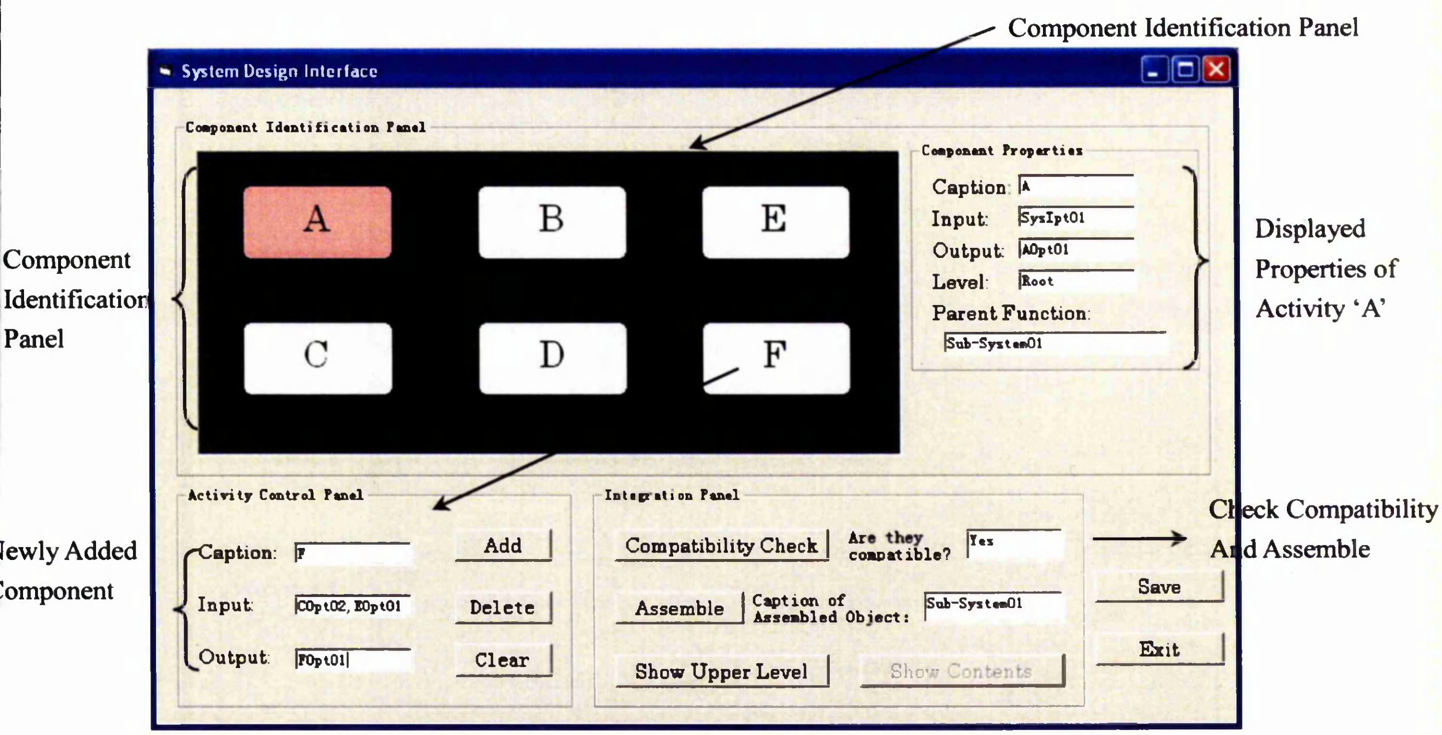


Figure 5.5: Root Level of the Graphic User Interface developed in Visual Basic

The identified activities can be checked as to whether they can perform a function jointly, by clicking the 'Compatibility Check' button. If the result is positive, the activities can be assembled and the formed sub-system can be inspected by clicking 'Show Upper Level'.

The SDI treats the exploration process as building blocks; as clients and service providers add required activities to generate more sub-systems, potential service solutions emerge. It enables the service provider and client to implement the identification procedure precisely and efficiently and have an instantaneous perception of potential solutions.

Significantly, the SDI considerably facilitates verifying the compatibility of potential solutions. Behind the graphic representation of each function or activity, its inputs and outputs have been defined. The compatibility check can be automatically implemented by running a program checking whether the required inputs of all select elements can find matched outputs.

A full description of the SDI is given in Appendix A, where the source code of the system compatibility check is presented.

5.4 Advantages and Disadvantages

The 'bottom up' process is an approach which enables FP providers to identify the necessary functions and activities to deliver a specified service. It is especially useful when a completely new system that is not derived from an existing service is to be implemented. The 'bottom up' approach also provides clients with a large degree of freedom to build the service system that they deem effective and efficient, through a high level of service customization.

The main disadvantage of the 'bottom up' approach is that it takes a comparatively long time to identify necessary components. A highly customized system may easily deviate from the experience of providers; hence, the business risk is rather high. Another disadvantage is that, when designing an extremely complicated SSS, the approach may result in a tangle of elements and sub-systems.

5.5 Summary

The 'bottom up' approach follows the principle that the individual base elements of the system are first specified in great detail, and are then linked together to form larger sub-systems, which then in turn are linked, sometimes through many levels, until the system eventually grows to the required level of complexity and completeness.

The approach provides a less restricted environment embracing both clients and providers to generate an appropriate SSS. In such a 'bottom up' process, compatibility checking of identified activities is required. Also, it may bring undesired complexity. A graphic user interface has been developed to visualize the activity selection procedure and facilitate the system compatibility check.

The main disadvantage of the 'bottom up' approach is that, although it offers clients a large degree of freedom to build highly customized and innovative SSSs, it is time consuming and has a high business risk.

Chapter Six

'Top Down' Service Design

6.1 'Top Down' Service Design Process

Based on the research and close investigation of SSSs in the food production and aerospace industries, a Generic Service Support System (GSSS) has been developed to be the departure point for the 'top down' design process. The 'top-down' process is so named as it starts from an information-rich format, that is, the GSSS which encompasses functions (activities) that could be generally applied to most cases. In addition, some functions could be further extended to more detailed (lower) levels according to client requirements. The identification of needed components is actualized by removing unwanted ones from the GSSS.

The 'top down' design process consists of the following steps:

Step 1: on receiving requirements from clients, the FP provider starts the design process with the GSSS;

Step 2: considering the system at the highest level, the client decides which functions and activities of the system are required;

Step 3: unwanted components are removed from the GSSS and a compatibility check is made on the remaining elements;

Step 4: the remaining system is then available for evaluation by the client.

In the 'top down' design process, the framework of the target SSS has been created prior to the beginning of the process. It is also possible to add further activities or decompose a function into the client's prescribed level of detail. During the process, providers and clients work interactively in an iterative manner; see figure 6.1.

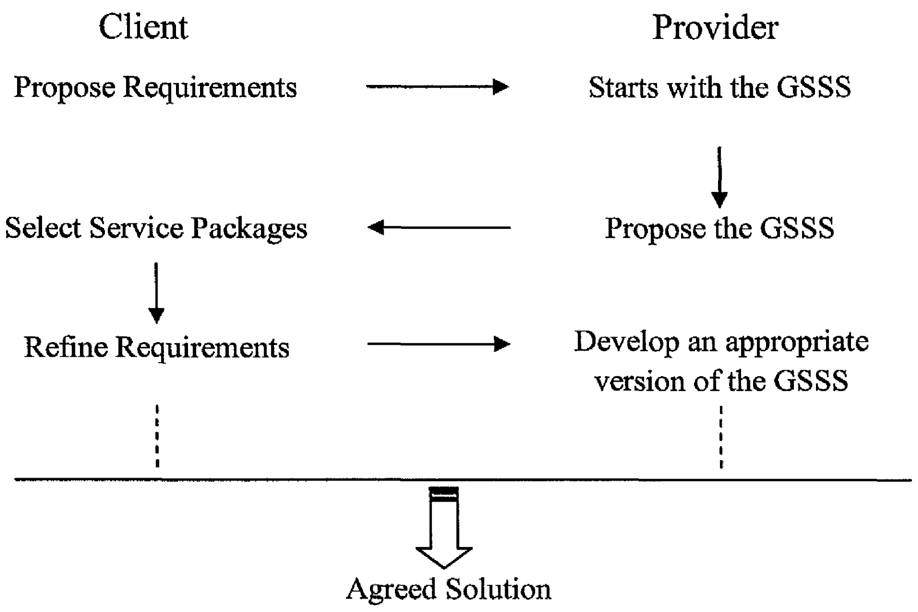


Figure 6.1: Iterative process of concept creation

The prerequisite and key to the 'top down' approach is the GSSS. The rest of this chapter elaborates on the GSSS by explaining each sub-system and activity. To illustrate its practicality, SSSs of two real FPs are compared with the GSSS. The chapter concludes with a comparison of the 'top down' and 'bottom up' service design approaches.

6.2 Generic Service Support System (GSSS)

The Generic Service Support System (GSSS) was developed from the findings of field investigations of real engineering support systems. The GSSS forms a standard version of product service system which could be extended or reduced according to specific requirements.

6.2.1 Top Level of the GSSS

The GSSS contains three levels, of which the top one encompasses five principal functions which are further decomposed into lower levels of detailed sub-functions and activities; see figure 6.2. The top-level description denotes external inputs and outputs of the GSSS (from client and partners) and the information and hardware flow within the service. Sub-functions and individual activities are described in the 2nd and 3rd levels. Table 6.1 describes the five principal functions.

Principal Function	Description
Customer Interface	Responsible for dealing with customer requests, assigning tasks
Activity Planning	Collecting all information, based on which the correspondent solution is tailored
Routine Maintenance & Repair	Implement actual maintenance or repair on prescribed HW
Hardware Procurement	Check hardware inventory, order and dispatch required HW
Database	Store all the information and deal with all enquiries

Table 6.1: Principal Functions of GSSS

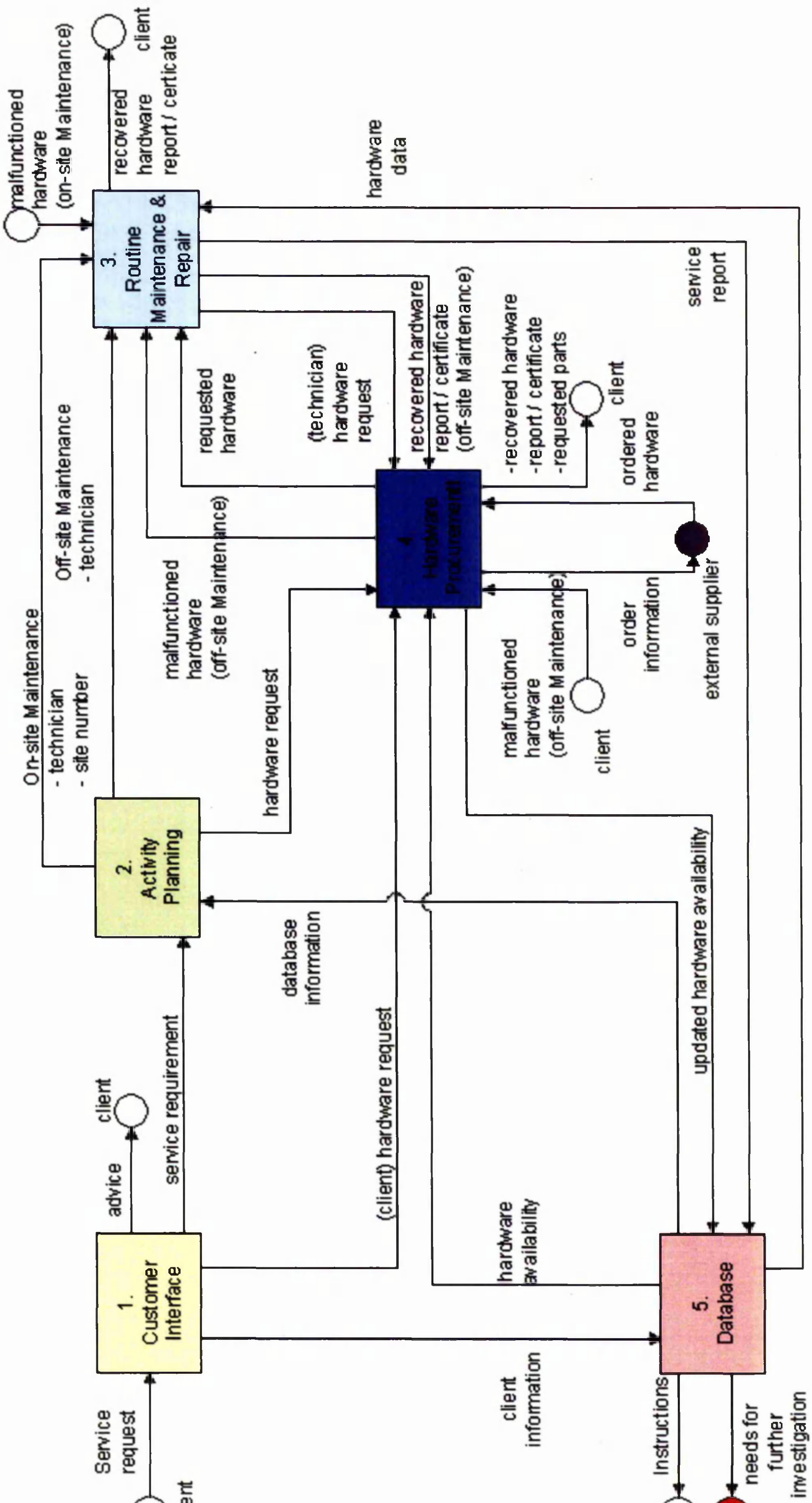


Figure 6.2: Top-Level Descriptions of GSSS

The output 'Need for further investigation' represents a situation in which the current system does not have the capability to solve the problems facing clients. Redesign of hardware may be implemented.

6.2.2 Customer Interface

The Customer Interface mainly deals with requests from clients. Such requests may include repair of a malfunctioning product, routine maintenance, technical enquires and requirement for parts. To respond, *Collect Information from Client* and *Analyse Request* are carried out; see figure 6.3. All client requests are systematically recorded in *Collect Information from Client* and transferred to *Analyse Request* where the appropriate service type is determined.

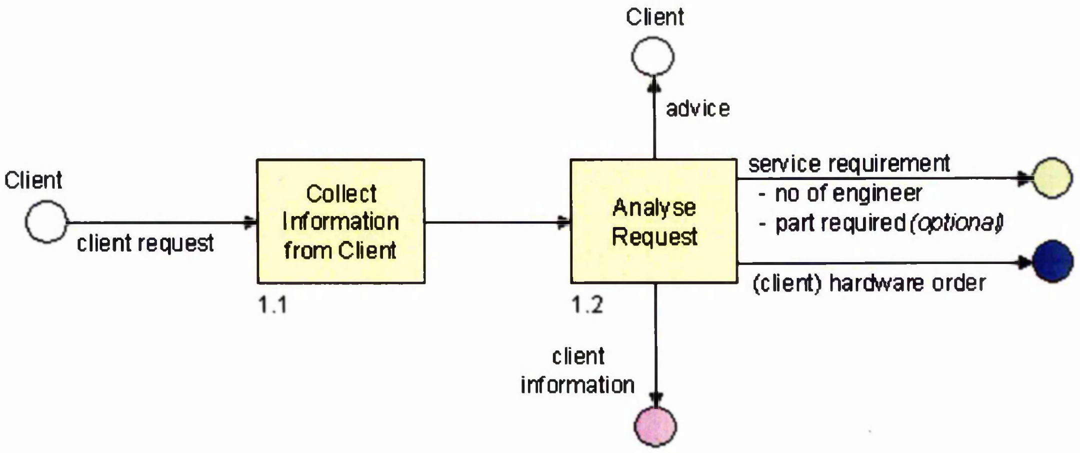


Figure 6.3: Customer Interface

On responding to enquires from clients, instructions or advice will be offered. The 'client information' shown in the figure 6.3 generally represents possible outputs such as further enquiries, unknown hardware failures and client profile information. These enquiries will be transferred to the 'database' where manuals are printed and delivered, inventory information of requested spare parts is checked, and client profile information is accessed and updated.

6.2.3 Activity Planning

Activity Planning results in the operational plan of a comprehensive service which is contracted to be implemented by FP providers or too complicated to be handled by clients. Three activities, *collect information, analyze information* and *make decision*, are carried out; see figure 6.4.

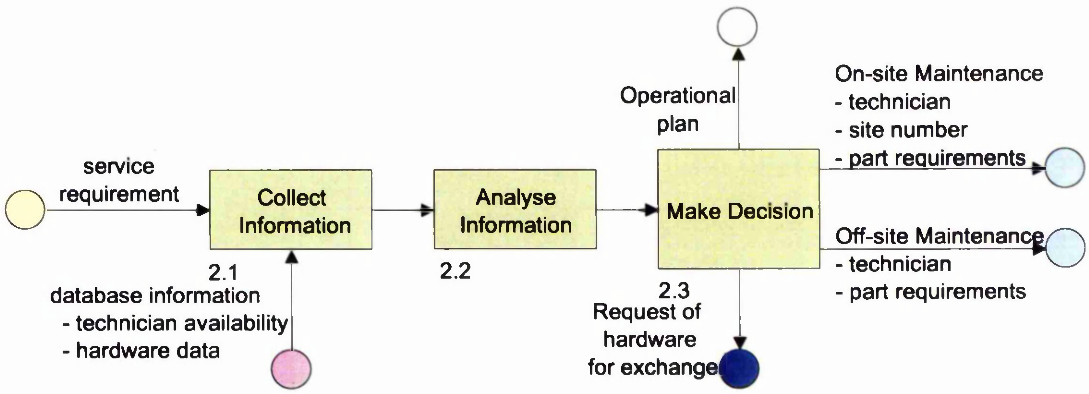


Figure 6.4: Activity Planning

Information to be collected includes requirements of clients, schedule of technicians, client-planned uses of hardware, and historical operation data of the specified products. Based on the analysis of collected information, decisions can be made on the maintenance type: on-site maintenance or off-site maintenance; and on the plan of the operations. In both cases, an operation plan is sent to the client specifying when to make the malfunctioning hardware ready, should on-site actions be implemented, or when hardware needs to be transported to the provider's base in the case of off-site maintenance.

6.2.4 Routine Maintenance and Repair

Routine Maintenance and Repair is a sub-system comprising main actions on hardware. It can be extended to the two lower-level sub-systems: on-site maintenance and off-site maintenance.

6.2.4.1 On-Site Maintenance

On-site maintenance comprises two components: 'Travelling to site' and 'Carry out on-site maintenance', which itself comprises more detailed activities; see figures 6.5 and 6.6.

On arriving at the client's site, the technician needs to check the database to acquire the hardware data; carry out an initial inspection; and liaise with personnel who are responsible for monitoring the hardware's condition. After these three activities are completed, a decision is made concerning what needs to be done to the hardware. The generated solution may include replacement of worn-out parts, in which case new spares are required. The resultant hardware request will trigger the activities in the 'Hardware Procurement' sub-system which is responsible for acquiring and transporting designated spares to the technician. A description of the 'Hardware Procurement' sub-system will be given in the next section.

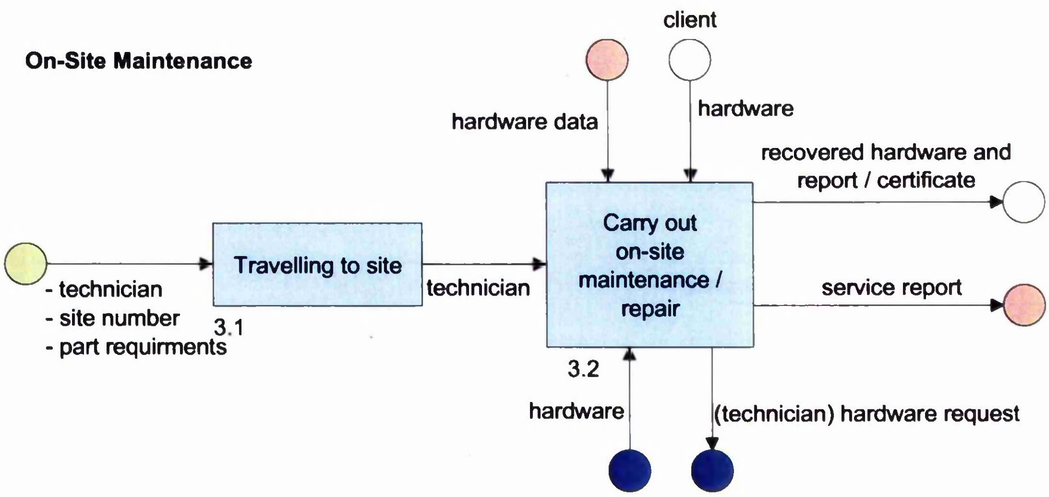


Figure 6.5: Components of Sub-system 'On-Site Maintenance'

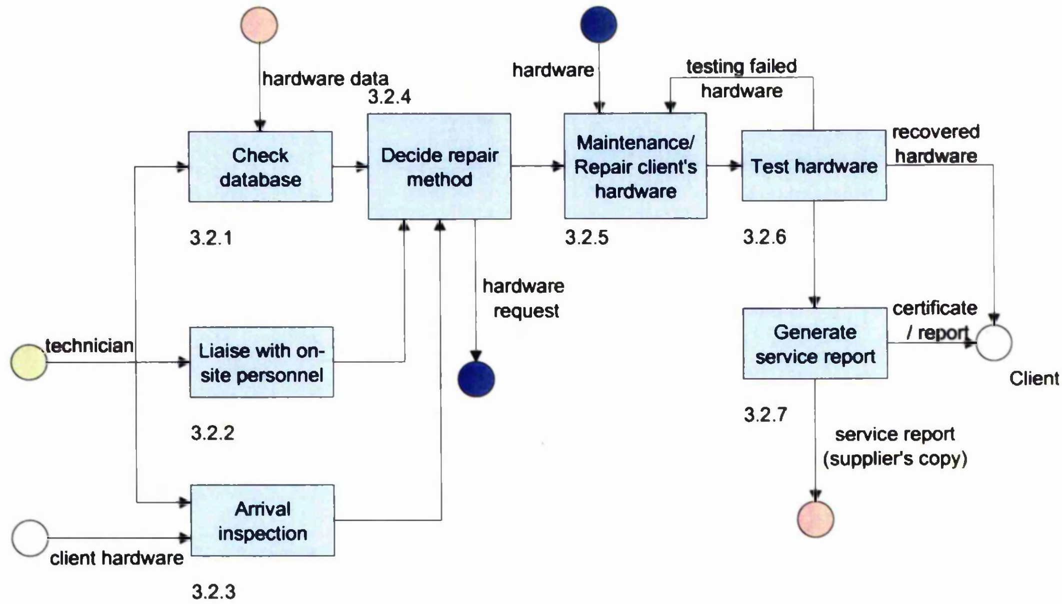


Figure 6.6: Description of Activities Comprised in 'Carry out On-Site Maintenance'

On receipt of the spares, actions on the hardware are implemented. This is followed by functionality testing and generating a service report. Usually a copy of the service report is stored in the FP supplier's database.

6.2.4.2 Off-Site Maintenance

Carrying out off-site maintenance occurs when the malfunctioning hardware must be transported to the provider's workshop where off-site actions are implemented; see figure 6.7. Transportation of hardware is a sub-function of 'Hardware Procurement'.

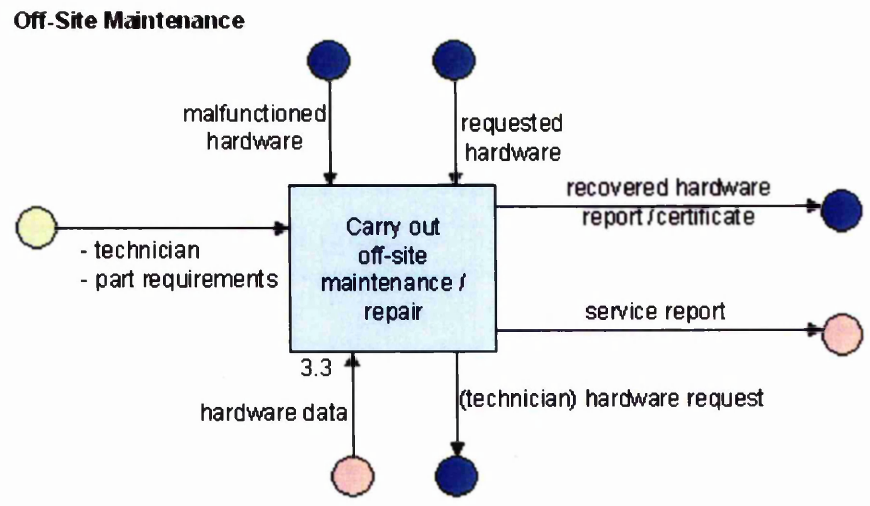


Figure 6.7: Off-Site Maintenance

Activities to implement an off-site service are similar to those for an on-site service; see figure 6.8.

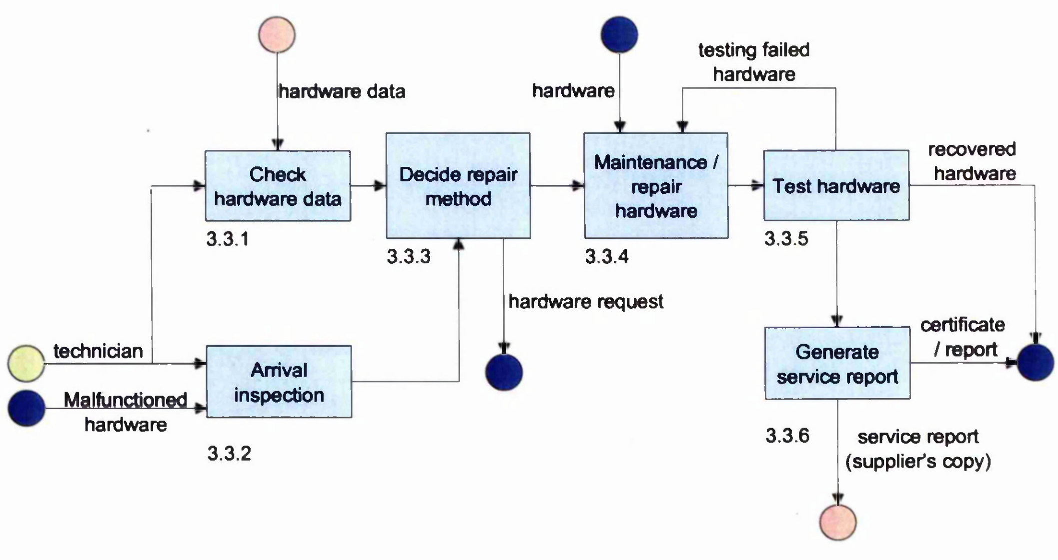


Figure 6.8: Third Level of the Routine Maintenance & Repair

6.2.5 Hardware Procurement

As previously mentioned, 'Hardware Procurement' is a function that acquires, stores and transports requested backup hardware or parts to the provider's workshop or client's site; see figure 6.9.

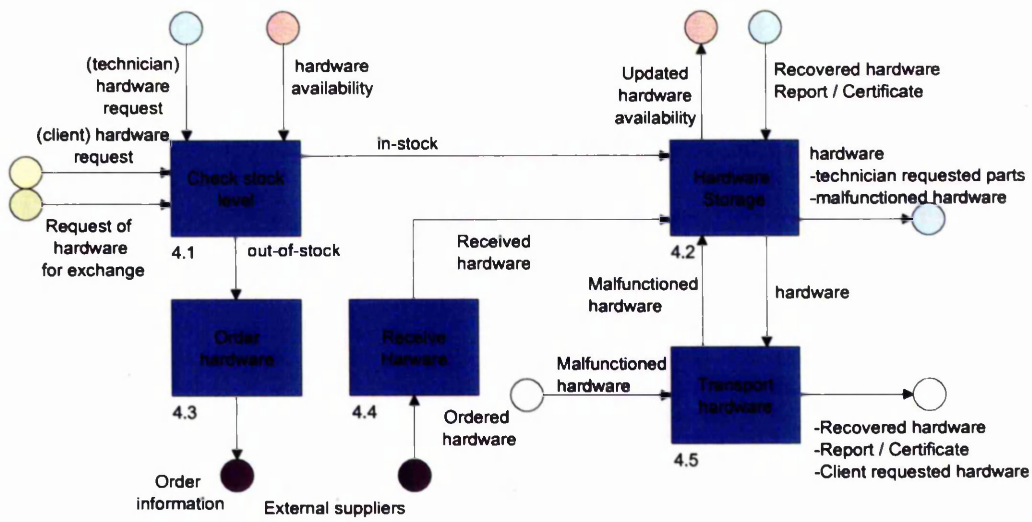


Figure 6.9: Second Level of Hardware Procurement

On receipt of a request for hardware, the first step is to check the stock level and decide whether it is necessary to order from external OEMs; this depends on the hardware availability information sent by the 'Database' function. If the requested hardware is in-stock, the next step is despatch from the warehouse. If requested hardware needs to be ordered, the lead time may be crucial for the some services that need to be completed on time.

'Transport hardware' contains two independent activities that deliver hardware to and from the client's site and the provider's base; see figure 6.10.

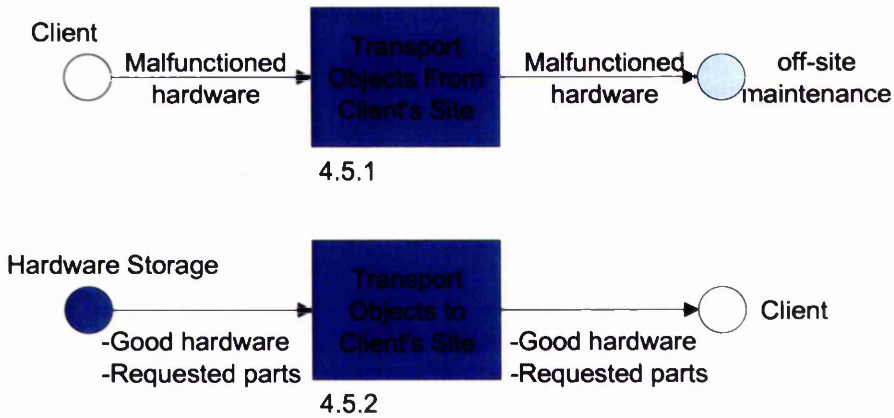


Figure 6.10: Third Level of Hardware Procurement

6.2.6 Database

The database is a sub-system that deals with all data enquires including hardware availability, technician schedule, hardware manual, etc. It also needs to store and update records of hardware that has been serviced; see figure 6.11. As previously pointed out, unknown hardware failures (failures that did happen before) need to be recorded and details sent for further investigation.

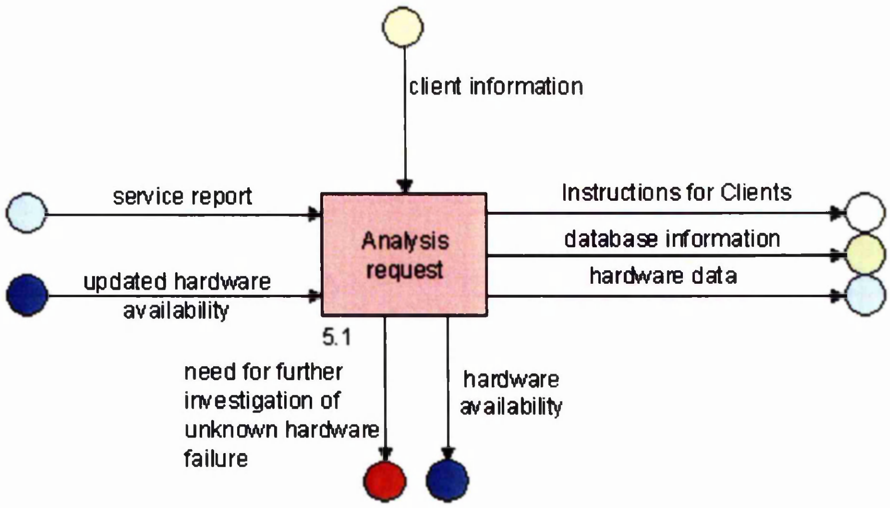


Figure 6.11: Database

6.3 Comparison with Industrial Examples of SSSs

The GSSS aims to be the template for service design using the 'top down' approach. To illustrate that the components comprised in the GSSS are applicable in real situations, comparisons between the GSSS and two SSSs from current FPs are given in this section.

6.3.1 Comparison with the SSS of Company A

Company A provides equipment and a service support system for manufacturers in the food industry. This case study concerns only the service support system. To maintain in good condition all machinery provided to the client company, this company designed a set of comprehensive support services of which routine maintenance is the theme, with repair in the event of machine breakdown. Company A is also responsible for recruiting, training and installing competent engineers. In all cases, company A agrees to supply an engineer as the best means of fulfilling the obligations under agreement.

Figure 6.12 illustrates the service operations in the SSS of company A. From the 'Blue print', a system model for company A was derived as shown in figure 6.13. The equivalent activities of company A's service system and the GSSS are shown in table 6.2.

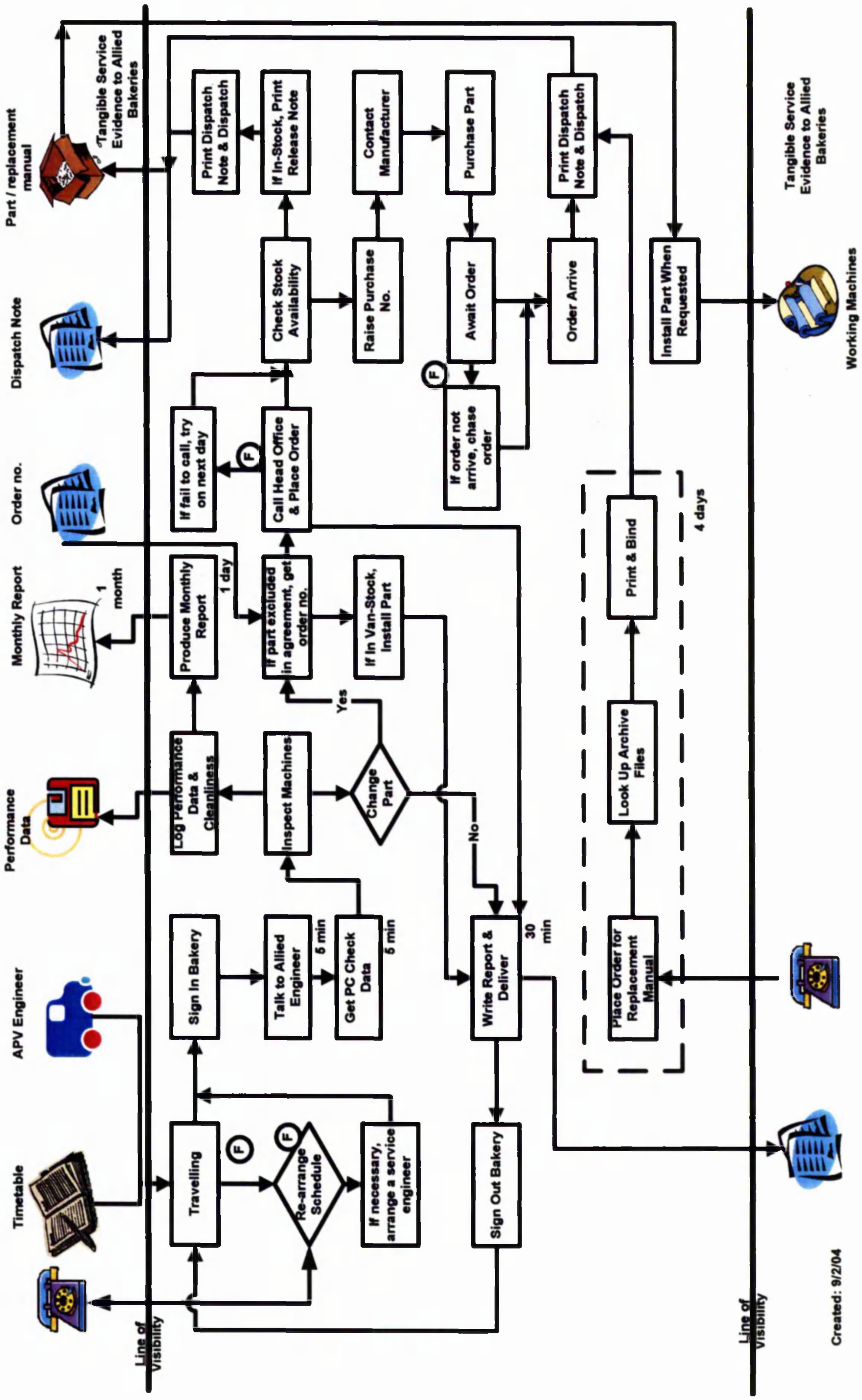


Figure 6.12: 'Blue print' of the service operations of Company A

Activities of the company A's system	Equivalent activities in the GSSS
Travelling Sign In Bakery Sign Out Bakery	Travelling to Site
If Re-arrange Schedule, Arrange a service engineer	Make Decision (Arrange an engineer)
Talk to Allied Engineer	Liaise with on-site personnel
Get PC, Check Data	Check Database
Inspect Machines	Arrival Inspection
Decide Change Part or Not Check if part excluded in agreement, Get order no. of the part to be changed	Decide repair method (Decide changing part or not)
If in Van-Stock, Install part	Maintenance/Repair hardware
Call Head Office & Place Order (if fail to call, try on next day)	Decide repair method (if changing part is required, generate a hardware request of technician)
Check Stock Availability	Check stock level
(If in-stock) Print release note	Hardware Storage
Print Dispatch Note & Dispatch	Transport hardware to Client's site
(if not in stock) Raise purchase no. Contact Manufacturer Purchase Part Await Order (Chase order)	Order hardware
Install Part When Requested	Maintenance/Repair hardware
Place Order for Replacement Manual	Analyze Request (Customer Interface, generating further request to Database)
Look Up Archive Files Print & Bind	Request Analysis (Database, generate instructions for Clients)
Log Performance Data & Cleanliness	Request Analysis (Database, save service report)
Produce Monthly Report Write Report & Deliver	Generate Service Report

Table 6.2: Equivalent Activities of Company A in the GSSS

Thus, the system model of company A's SSS can be obtained by removing the remaining activities from the GSSS. The resulting service system contains five main functions that are same as the GSSS; see figure 6.13.

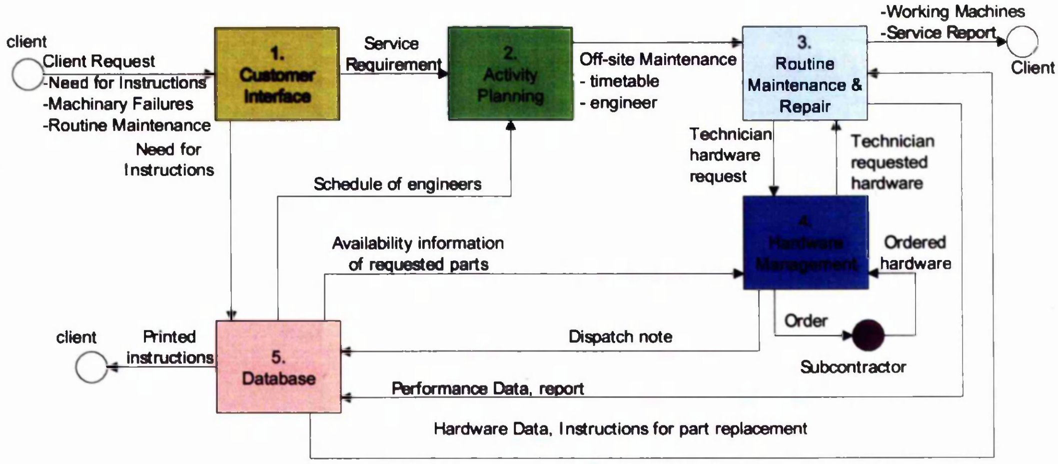


Figure 6.13: Top Level Description of the SSS of Company A

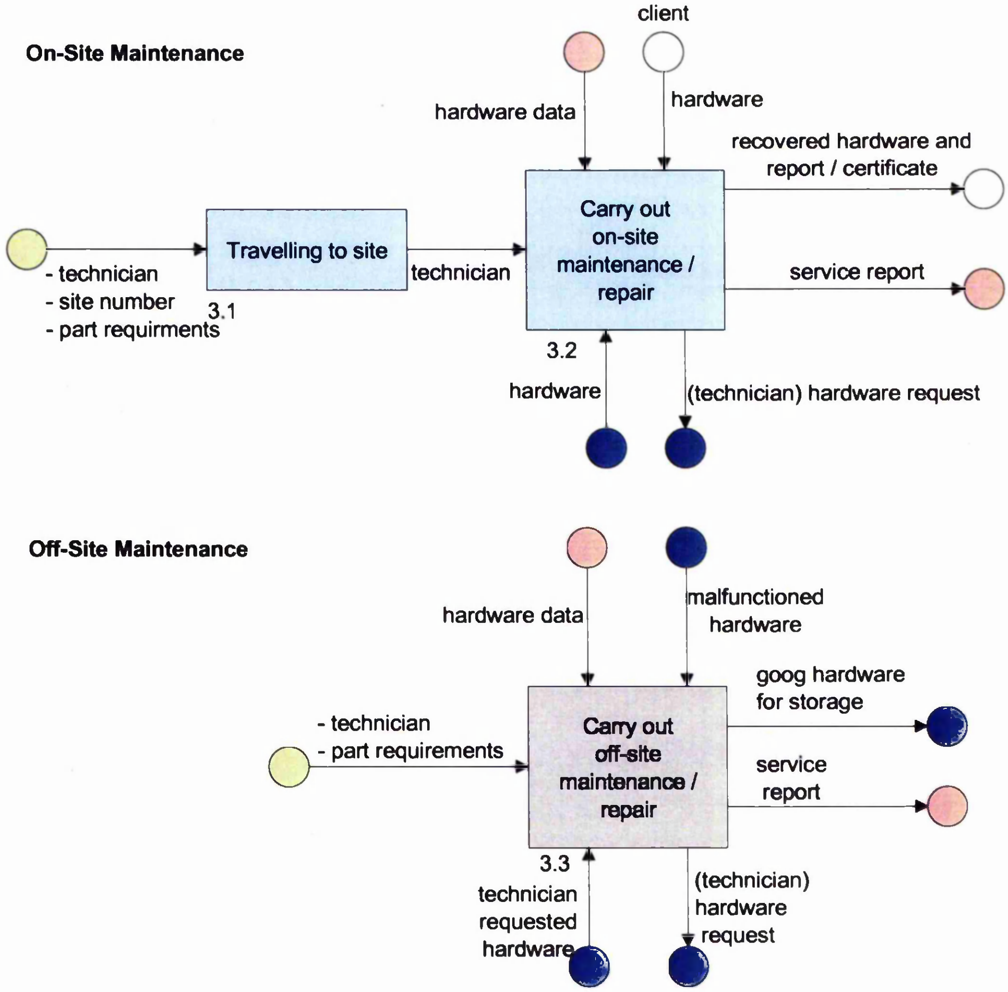


Figure 6.14: Second Level Description of Routine Maintenance & Repair

Figure 6.14 shows the system model obtained from the GSSS. Company A's SSS is designed to be on-site maintenance only; therefore, such a change could be made by removing the off-site maintenance sub-function from the GSSS (boxes in grey). Another example is shown in figure 6.15. Company A does not specify a testing activity in its service operations; thus, by removing 'Test machines', the remaining activities form a sub-system meeting the company's requirements. Without 'Test machines', the outputs from 'Carry out actions on client's machines' are sent directly to 'Generate service report' and the client; therefore, the result is compatible.

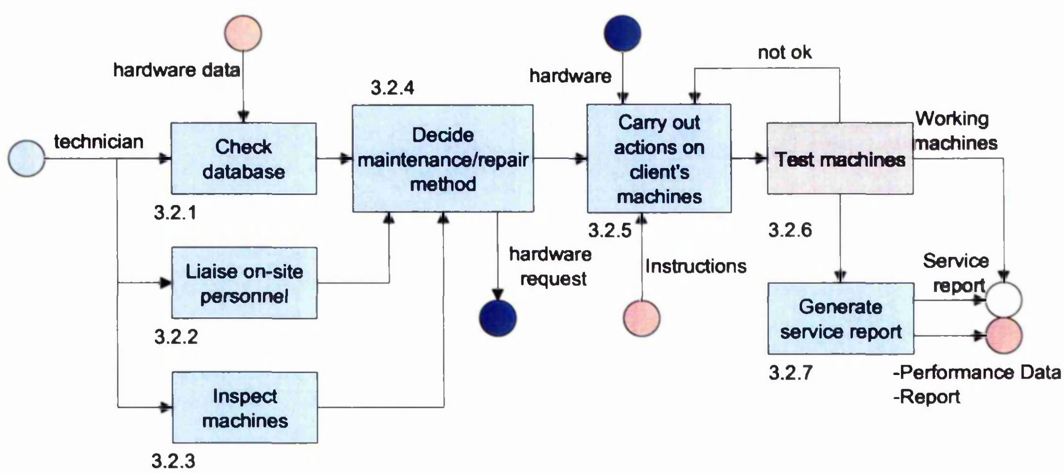


Figure 6.15: Third Level Description of Routine Maintenance & Repair

6.3.2 Comparison with the SSS of Company B

Company B has one of the world's largest independent maintenance, repair and overhaul centres for aircraft engines. Company B has established a SSS to assure the availability of their products. As aircraft engines are safety-critical products and very complex, any problem with them needs to be inspected comprehensively. In some cases, it is not possible to repair an engine on the wing and remanufacture is required. Such overhauls take weeks to carry out. The policy of company B to guarantee the promised availability is implemented by an engine exchange system. The repaired malfunctioning engine will be stored in the service base for future change.

The SSS of company B can also be obtained by removing unwanted elements from the GSSS. A detailed comparison between the service blueprint and the SSS obtained from the GSSS will be given in chapter 8, where a case study of applying the 'top down' design approach is carried out and a comprehensive performance evaluation of the SSS with respect to its reliability is presented.

6.4 Comparison with the 'Bottom Up' Design Process

The 'top down' approach provides a rapid process for SSS design. Given a set of client requirements, an appropriate version of the system can be generated from the GSSS. A computer-based system compatibility check is not necessary as the changes made to the GSSS are usually not complicated. Iterative interactions between the service providers and clients provide a degree of system customization. Systems can be readily tested by running a pre-built simulation model; therefore, the design time is less than that of a 'bottom up' approach. Reduced design time can reduce the cost for all parties involved and more effort can be put into achieving a reliable system.

Another advantage compared with the 'bottom up' method is that the 'top down' method generates service solutions based on a template with which FP providers are familiar; therefore it incurs less business risk.

The GSSS allows customers who are not familiar with service system design to create a system to meet their needs. The 'bottom up' approach does not provide the basic infrastructure for customers to build their system from a 'blank sheet of paper'.

6.5 Conclusion

The 'top down' approach is a rapid design process for SSS developments. In the process, the GSSS plays a key role and serves as the departure point. It comprises most useful activities that could constitute a SSS for general purposes. The design process brings less business risk compared with the 'bottom up' design method; also, it enables customers who do not have knowledge of how SSSs operate to acquire quick insight into how the SSS under development will work.

Chapter Seven

Design Support Tool

7.1 Introduction

The design process has created a network of activities that describe a service support system. The objective is now is to understand how such a system might fail and to create a design tool that helps the designer to model the system performance with respect to system failure.

A method of system modelling will satisfy design and failure modelling requirements. This chapter discusses the way in which a SSS might fail and shows how a system modelling method can be used to evaluate the performance of the SSS.

7.2 Failure Characteristics of SSSs

7.2.1 Contract Failures and Customer-Perceived Failures

The failure of SSS can be defined objectively in that there exists a contract that specifies the level of service to be provided by the SSS. The level of performance of Key Performance Indicators (KPIs) may be monitored. Typical kPIs include *time to undertake a service* and *cost.* Any fall below the contracted level of performance may be defined as a contract failure; see figure 7.1.

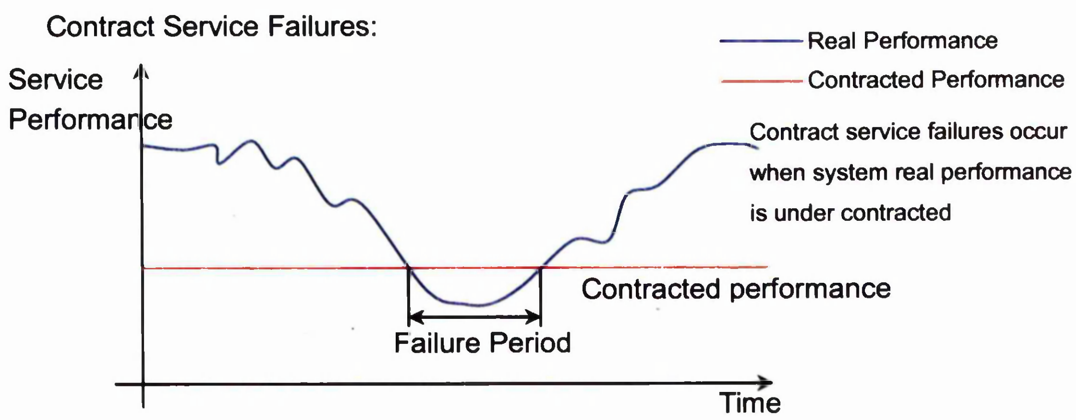


Figure 7.1: Contract Service Failures

However, customer-perceived failures may exist even when contracted failures do not. Should the actual performance be well above the contracted performance for long periods, then customer expectations are raised. Any fall in actual performance below the expectation of performance is then perceived as a failure, even though the actual performance is not below the contracted performance; see figure 7.2. Though customer-perceived failures do not incur financial penalties for the FP providers, they do harm the long-term relationship with customers.

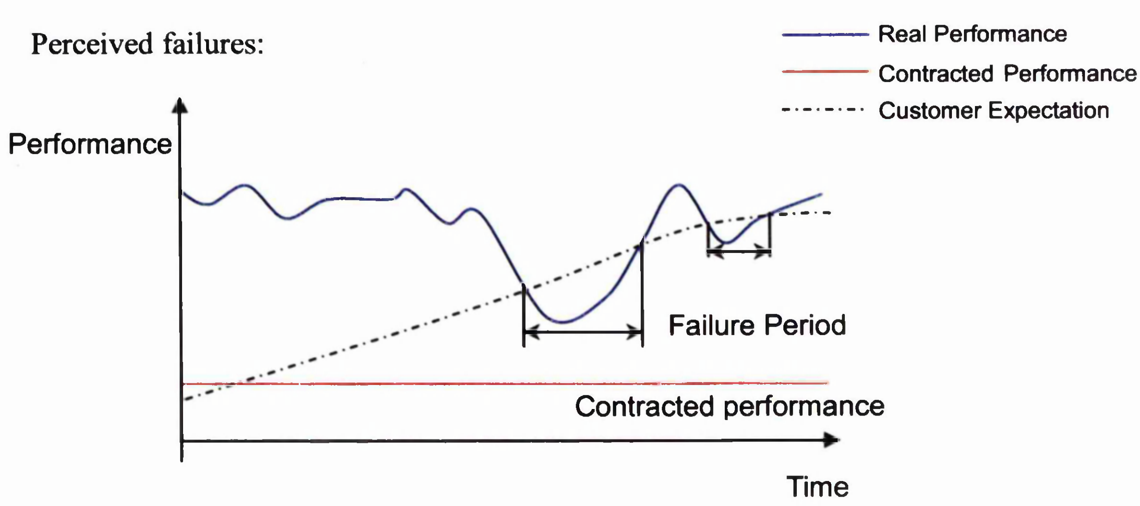


Figure 7.2: Customer-Perceived Failures

The optimum system performance with respect to the contract is one with low standard deviation and minimum 'safety margin' (vary according to the contracts) above the contracted performance; see figure 7.3. Thus customer expectation and satisfaction are managed well at minimum cost, i.e. resources are not used to generate performance well above that contracted.

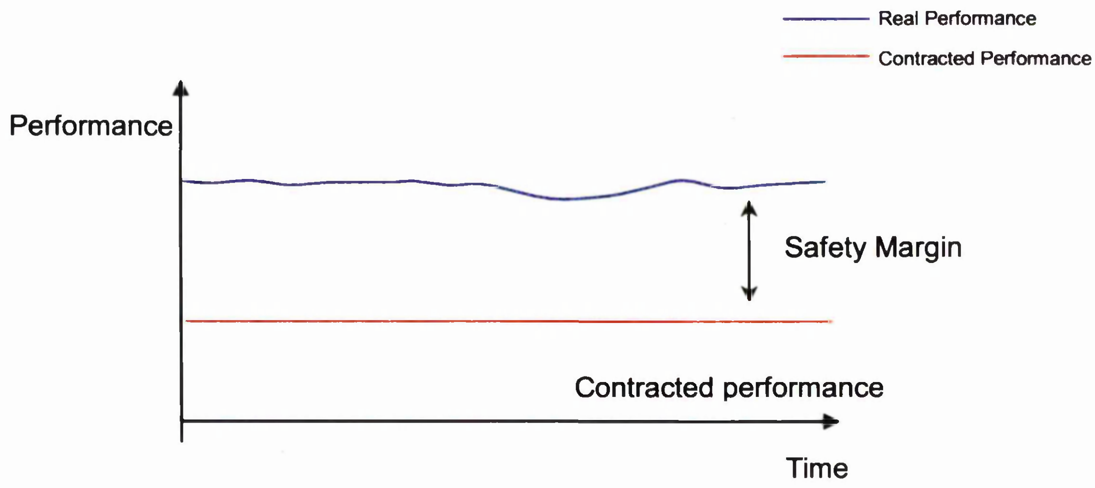


Figure 7.3: Stable System Performance

7.2.2 Internal Failure and External Failure

Failures of a SSS can be divided into two groups: internal and external. External failures are those that can be seen by customers and relate to service quality and expectations. Typical examples are low quality and late deliverables (contract failure), and perfunctory and slow reaction to client enquires (customer-perceived failure).

Internal failures (activity failures) relate to system inefficiency and individual events may not be evident to the customer. For example, the transfer of inaccurate data between activities may lead to repeat work and hence increased cost and time delay, but the customer may be unaware of the problem if the overall system performance is satisfactory.

An internal failure does not necessarily cause the system to fail; it only necessitates a repeat action which increases the time taken and the resources used to complete the system requirements satisfactorily. However, a number of internal failures can combine to create an external failure.

7.2.3 Critical Activities

The failure characteristics of SSSs lead naturally to a consideration during the design process of how internal failures may be reduced to avoid external failures (perceived or contractual). It is essential to explore the effect of resources and reliability of an activity on the system performance. Such sensitivity analysis can reveal the critical activities and give suggestions for re-allocation of available resources to improve the reliability of the SSS.

7.3 Reliability of SSSs

The functional performance of the SSS (with respect to a KPI) is the functional reliability of the system,

> *The probability that the system will not perform below a prescribed level, for a defined period of time, with specified resources and operating in a given environment.*

The overall system reliability is the % of time that the overall performance is well above the contracted level during a defined time period ('functional period/total time' in figure 7.1). During the monitored period, each performance of the service may be plotted; see figure 7.4. Intervals (not evenly distributed) between services do not represent the time the system is in residence in a functional or failed state because the SSS may be not be in use, or maying be carry out multiple service tasks simultaneously.

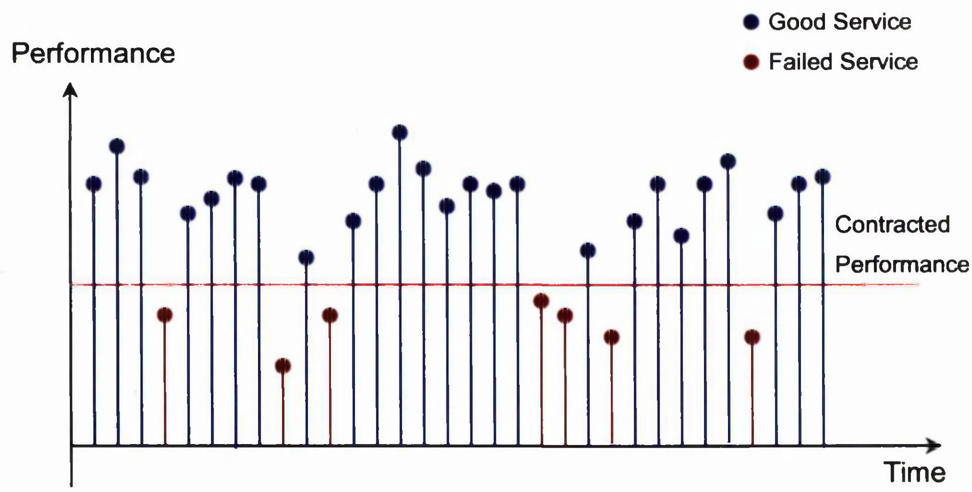


Figure 7.4: Performance Plots of Services

During a period of time, a large number of services may be delivered. Thus, during a defined period of time, the reliability of a SSS (***R*$_s$**) is calculated by:

$$R_S = \frac{\text{Total Number of Services - Number of Failed Services}}{\text{Total Number of Services}} \tag{7.1}$$

7.4 Modelling SSSs

A system model can be created in which the performance of individual activities is combined to give a total system performance. The performance of individual activities can highlight internal failures and the total system performance will show any external failures (perceived or contractual) in a SSS model. The principal functions are modelled by particular activities which are connected by their respective inputs and outputs in a system network.

7.4.1 Activity Modelling

An activity is described by the time taken, the resources consumed and its input(s) and output(s). Table 7.1 lists parameters used to model an activity.

Parameters	Description
Activity Time (t_a)	The time taken to carry out the activity, which follows the normal distribution and is denoted by T_m and T_d
Accumulated Time (T_a)	The total time taken (including carrying out repeated activities if needed) to generate satisfactory output(s)
Resources (C_a)	The resources required (person-hours)
Cost (P_a)	Total resource consumed
Reliability(R_a)	The 'reliability' of the activity

Table 7.1: Parameters of an Activity

The reliability (R_a) of an activity is a measure of the ability to deliver correct output(s). If, say, R_a =0.9, then over a long time 10% of the output of this activity is not satisfactory and, in these 10% of cases, the activity must be repeated, as shown in the 7.5.

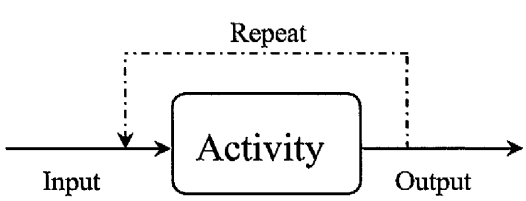


Figure 7.5: Potential Failure of Activity with Single Input and Output

When modelling, a simple method is to generate a random number (N_{rnd}) between 0 and 1 at the end of each activity. Using the case of R_a = 0.9, if N_{rnd} > 0.9 then the activity is repeated. In more complex models, incorrect information may be acted on by subsequent activities and a whole set of activities repeated when the error is discovered at a later time; this, obviously, will lead to considerable increase in the whole service time and cost, two important KPIs.

When one activity has been executed, another activity receives an input and generates outputs, as listed in table 7.2.

Inputs	Outputs
Required input(s) of the activity	Output(s) generated by the activity
	Total time taken
	Total resource consumed

Table 7.2: Inputs and outputs of an activity

The time taken to perform an activity is calculated as follows:

The mean activity time is affected by the resources available to implement the activity. Assume that $\boldsymbol{T_m}$ is in inverse proportion to $\boldsymbol{C_a}$. If $\boldsymbol{T_m}$ is the mean time taken for a defined unit resource, then:

$$T_{m,actual} = T_m(C_a)/(C_{a,actual}) \tag{7.2}$$

Where $\boldsymbol{C_{a,actual}}$ can be expressed as a % increase/decrease on $\boldsymbol{C_a}$ or a quantity of resource.

$\boldsymbol{t_a}$ is the time taken each time the activity takes place, whether the output is good or bad. $\boldsymbol{t_a}$ is sampled randomly from a normal distribution of mean $\boldsymbol{T_{m,actual}}$ and standard deviation $\boldsymbol{T_d}$ each time the activity takes place. Therefore the accumulated time taken is:

$$T_a = n \times t_a \tag{7.3}$$

Where $\boldsymbol{n}$ is the total number of times an activity generates output(s).

The cost of an activity is proportional to the resources used. The resources used for an activity are:

$$P_a = n \times C_{a,actual} \quad (7.4)$$

7.4.2 Activity Simulation

MATLAB®/Stateflow was used for modelling and simulation. Figure 7.6 illustrates an example of the simulation of activity 'A' to show how a single activity is modelled.

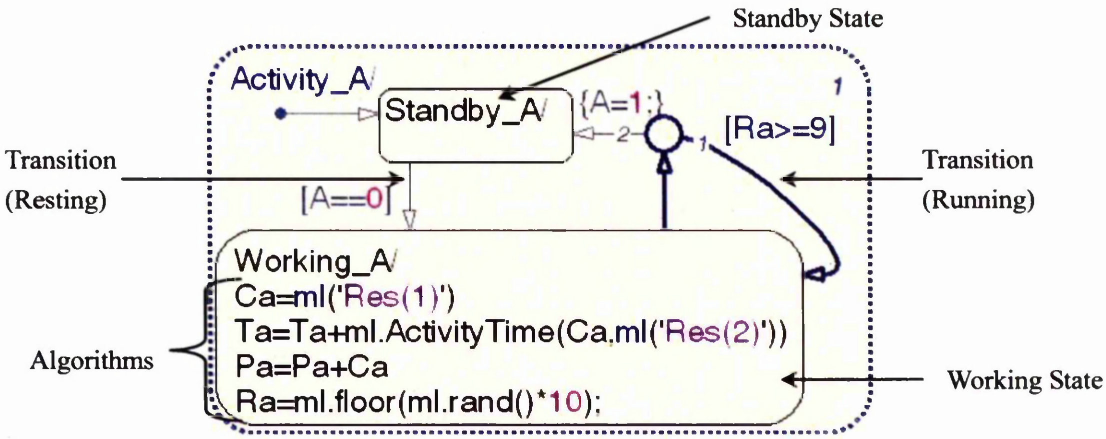


Figure 7.6: The Simulation of a Single Activity in Stateflow

Stateflow charts enable the graphical representation of the activities and comprise three main parts: states, transitions and algorithms; see figure 7.6. An activity is modelled with two states, working and standby, between which transitions are incurred by receiving the activity input(s) and generating output(s). Unqualified output(s) bring the activity back to the working state; that is, repeating the activity.

With reference to figure 7.6:

1) Activity_A is the name of the activity;
2) Standby_A is the standby state of the activity;

3) Working_A is the working state of the activity;
4) [A==0] is the condition which has to be met to transit to the working state;
5) C_a is the resource (persons) available for each activity; **ml('Res(1)')** is the value for C_a;
6) **ml.ActivityTime** (T_m, T_d)

   t_a is sampled randomly from a normal distribution of mean T_m and standard deviation T_d each time the activity takes place.

   '**ml('Res(2)')**' is the value for T_d.
7) **Ta=Ta + ml.ActivityTime** (T_m, T_d)

   This function calculates the total time taken by each activity during a service. An activity may fail during the service, and repeating the activity leads to an accumulation of the time consumed.
8) **Pa=Pa + Ca**

   This function calculates the total resources consumed by each activity during a service. **Pa** will accumulate if an activity repeats.
9) **Ra=ml.floor(ml.rand()*10)**

   The function '**ml.floor(ml.rand()*10)**' generates a random number (N_{rnd}) between 0 and 10 and assigns it to '**Ra**' at the end of each activity.
10) **[Ra>=9]**

    It designates the condition that the activity will repeat. If the system reliability is 0.9, [**Ra** > 9] will lead the activity to repeat.
11) **{A=1;}**

    The output of the activity A.

7.5 Design Support Tool for System Performance/Failure Modelling

Activities in a SSS can be categorized by their inputs and outputs into four basic types (see chapter 4). Integrating these four typical activities forms a basis for creating the model for a whole service system. The details of activity modelling are given in the Appendix B. Therefore, the total system model for the GSSS, described in chapter 6, can be created.

This model can:

1. support the 'top down' design method. Evaluation of the performance of the developed service system with the 'top down' approach could be implemented by running a reduced version of the simulation model of the GSSS;
2. model the system performance so as to highlight internal and external failures;
3. carry out sensitivity analysis so as to find critical activities with respect to system reliability.

Validation of the GSSS was implemented by running with the data from company A (see chapter 6) and comparing the simulation results with the recorded performance of the service system. During the validation, the simulations output the parameters given in table 7.3.

Parameters	Description
Administration Time (per day)	the time taken to do administration work, for example generate service report, on every working day
Administration Time (per service)	the time taken to do administration work during each service (a service may be carried out on several days)
Travel Time (per day)	the time taken to travel to the client's site (on each day)
Maintenance Time (per day)	the time taken to implement actions on hardware (on each day)
Total Administration Time	the total time consumed by doing the administration works
Total Travel Time	the total time consumed in traveling
Total Maintenance Time	the total time consumed by actions on hardware

Table 7.3: Parameters Output by the Simulation Model

The first simulation was performed for 39 days during which 35 service missions were carried out. The results of the simulation are compared to the company data in figures 7.7 and 7.8.

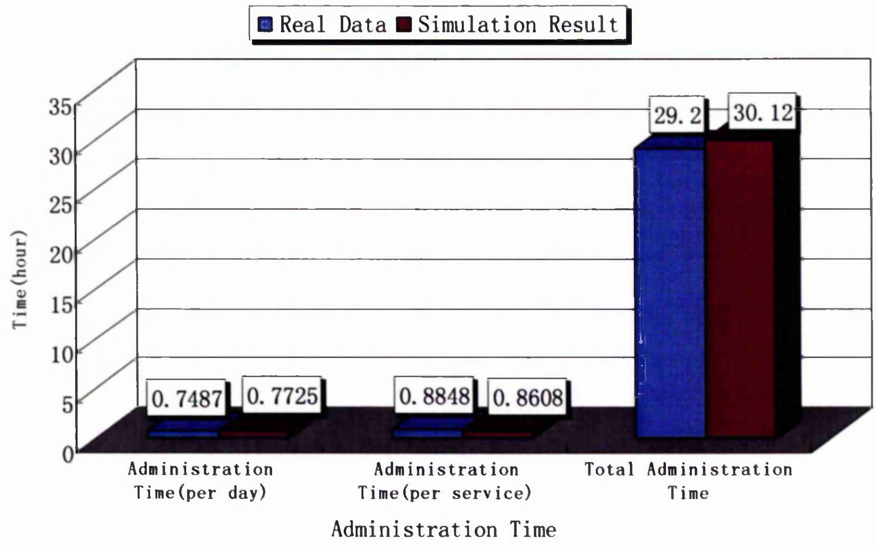


Figure 7.7: Comparison of Results: Administration Time (Simulation of Site 1)

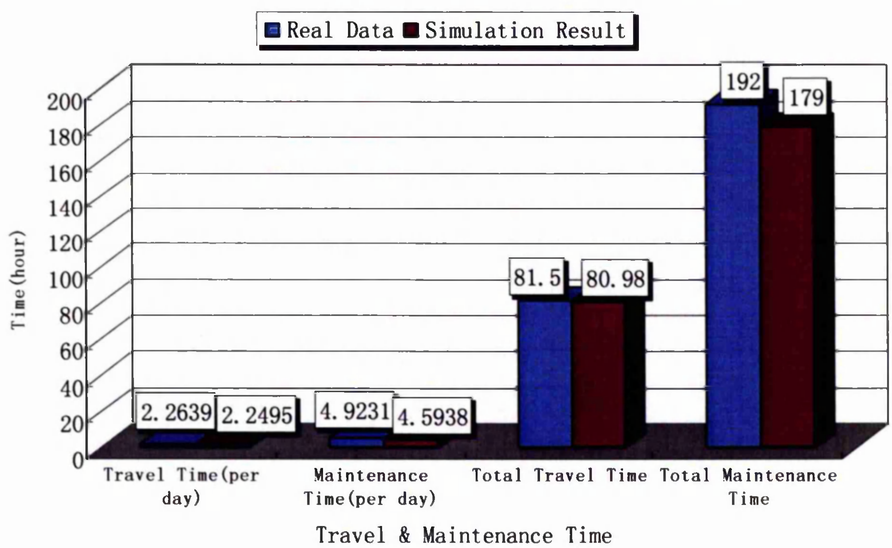


Figure 7.8: Comparison of Results: Travel and Maintenance Time (Simulation of Site 1)

It can be seen from figures 7.7 and 7.8 that simulated results compare well with the company data. Mean error for all parameters are given in table 7.4. The maximum error is in the parameter Total Maintenance Time and is 6.77%.

Parameters	Mean Errors
Administration time (per day)	3.08%
Administration Time (per service)	2.4%
Travel Time (per day)	0.6%
Maintenance Time (per day)	6.7%
Total Administration Time	3.05%
Total Travel Time	0.6%
Total Maintenance Time	6.77%

Table 7.4: Mean Errors of Parameters (Site 1)

The second simulation was performed for 54 days during which 35 service missions were carried out. The results of the simulation are compared to the company data in figures 7.9 and 7.10.

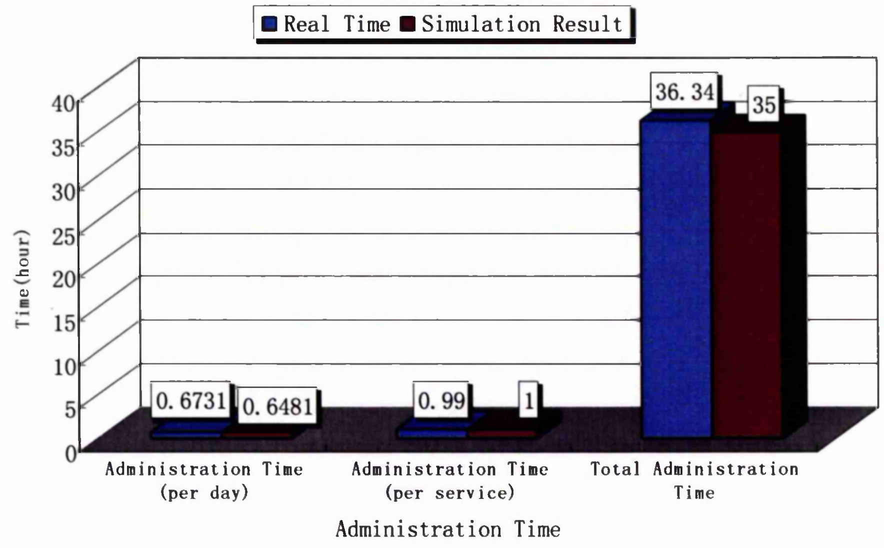


Figure 7.9: Comparison of Results: Administration Time (Simulation of Site 2)

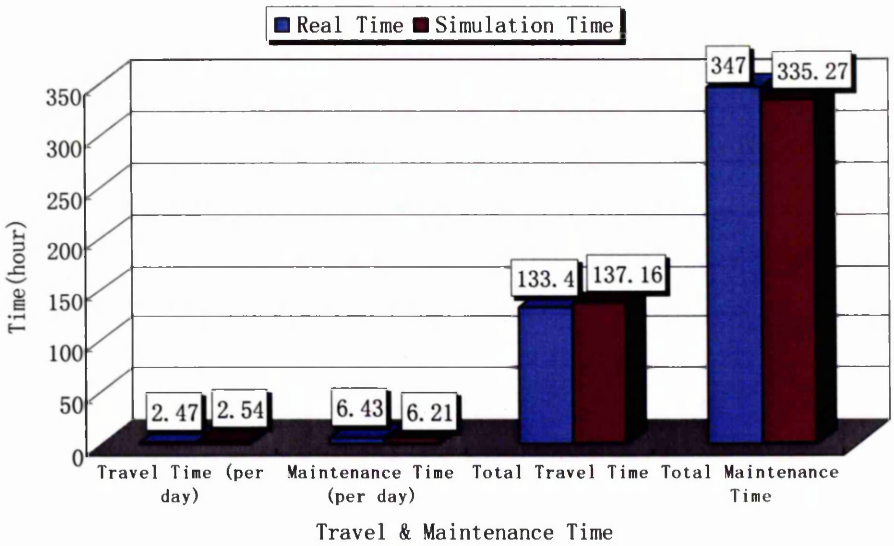


Figure 7.10: Comparison of Results: Travel and Maintenance Time (Simulation of Site 2)

It can be seen from figures 7.9 and 7.10 that simulated results of site 2 compare well with the company data. Mean errors for all parameters are given in the table 7.5. The maximum error is in the parameter Administration time (per day) and is 3.71%.

Parameters	Mean Errors
Administration time (per day)	3.71%
Administration Time (per service)	1%
Travel Time (per day)	2.76%
Maintenance Time (per day)	3.42%
Total Administration Time	3.68%
Total Travel Time	2.74%
Total Maintenance Time	3.38%

Table 7.5: Mean Errors of Parameters (Site 2)

Therefore, comparison of results shows that the simulation model can predict system performance with accuracy.

7.6 Failure Modelling of a Simple SSS

A pilot use of the system modelling method was applied to a simple SSS. In this section, the method will evaluate the reliability of the SSS, study the causal relationship between internal failure(s) and external failure(s) and find the critical activities with respect to the system reliability. A major case study is given in Chapter 8.

7.6.1 System Description

A simple system network consisting of four activities was analysed with respect to reliability; see figure 7.11. On receiving input $\boldsymbol{S_{Ipt}}$, the system starts to function, activity A is first triggered and generates two outputs, $\boldsymbol{A_{Opt01}}$ and $\boldsymbol{A_{Opt02}}$. These outputs are then sent to activity B and activity C as inputs respectively. Activity B and activity C work in a parallel manner and generate outputs $\boldsymbol{B_{Opt01}}$ and $\boldsymbol{C_{Opt01}}$, both of which have to be ready to trigger activity D where the system output $\boldsymbol{S_{Opt}}$ is generated.

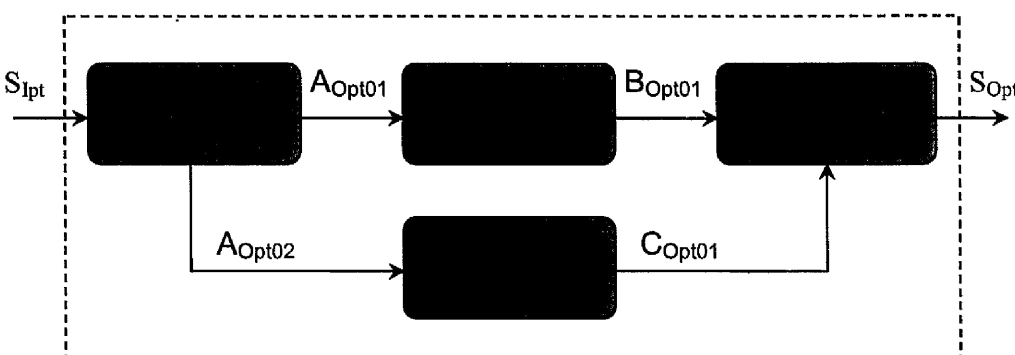


Figure 7.11: Sample SSS

The system performance can be evaluated through two parameters which are:

1) Time (T_S) – the total time taken by the SSS to generate the system

output;

2) Cost ($\boldsymbol{P_S}$) – the total resource consumed by the SSS.

Successful services delivered by the system are those generating the required output within the contracted time. Thus, a contract service failure can be defined as the service taking longer time than that prescribed in the agreements.

7.6.2 System Reliability of the Sample SSS

A simulation was run specifying the values of the parameters of each activity; see table 7.6. The units for resource and time are 'person' and 'hour' respectively. The reliabilities $\boldsymbol{R_a}$ of each activity are fixed during each simulation run and the default values of all activities are assumed to be 0.9.

Parameters / Activity	Resources (person) (C_a)	Activity Time(hour)		Reliability (R_a)
		Mean (T_m)	Deviation (T_d)	
Activity A	1	4	1	0.9
Activity B	1	1.5	0.5	0.9
Activity C	1	0.5	1/6	0.9
Activity D	1	2	1/3	0.9

Table 7.6: Values of Parameters of the SSS

It was assumed that the contract included 10 hours' work. Therefore, any service time taking longer than 10 hours is a failure.

The simulation was performed 100 times and the results are plotted in figure 7.12. (N.B. Failed services are plotted above the contract line in this figure.) In the simulation, 16 service failures were encountered, therefore the reliability of the SSS is:

$$R_S = \frac{100-16}{100} = 0.84$$

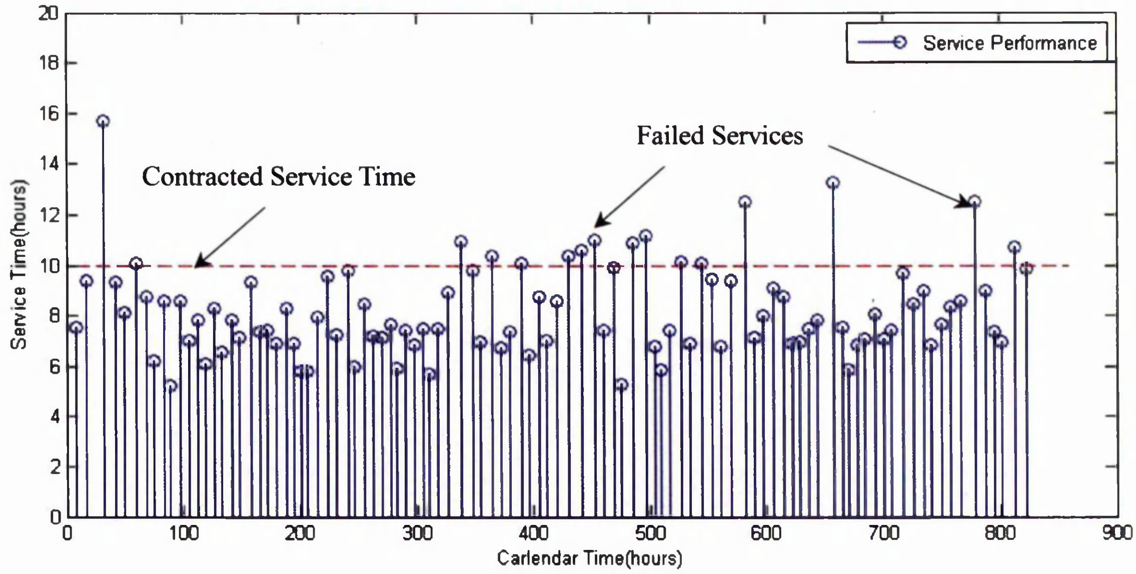


Figure 7.12: Simulation Results

The system presents a performance with a rather high risk of failing to deliver the service within the contracted time. Thus, there is a requirement for re-allocating or giving more resource to the system to reduce the time taken.

7.6.3 Causality between External and Internal Service Failures

Although some or some combinations of activity failures (internal failures) can result in external failures, there exists no explicit causal relationship between internal failures and external failures. Whilst the total system failure is caused by individual failure(s), the same activity may not always be the cause of the system failure.

This point can be illustrated by taking 12 examples from a set of 50 runs; see table 7.7. Table 7.7 lists the number of failures of each activity and the SSS during 12 services and it shows that there is no determined causality between internal and external failures. An activity failure alone may incur a system failure; however, sometimes, the failure of the same activity combined with another activity failure would not lead the system to fail.

For example, failure of activity B caused service 8 to perform badly, but it did not affect the service performance when combined with the failure of activity D in service 12. The abnormal phenomenon originates from the uncertainty of the activity performance; that is, the uncertain activity time taken on each occasion. The total time the system takes to generate a qualified output is accumulated as each activity executes. Failures of activity B lead it to repeat; however, the time taken on each occurrence could vary considerably; see figure 7.13. (Times of activity B follow a normal distribution with a mean 1.5 hour and a std. 0.5; see table 7.6)

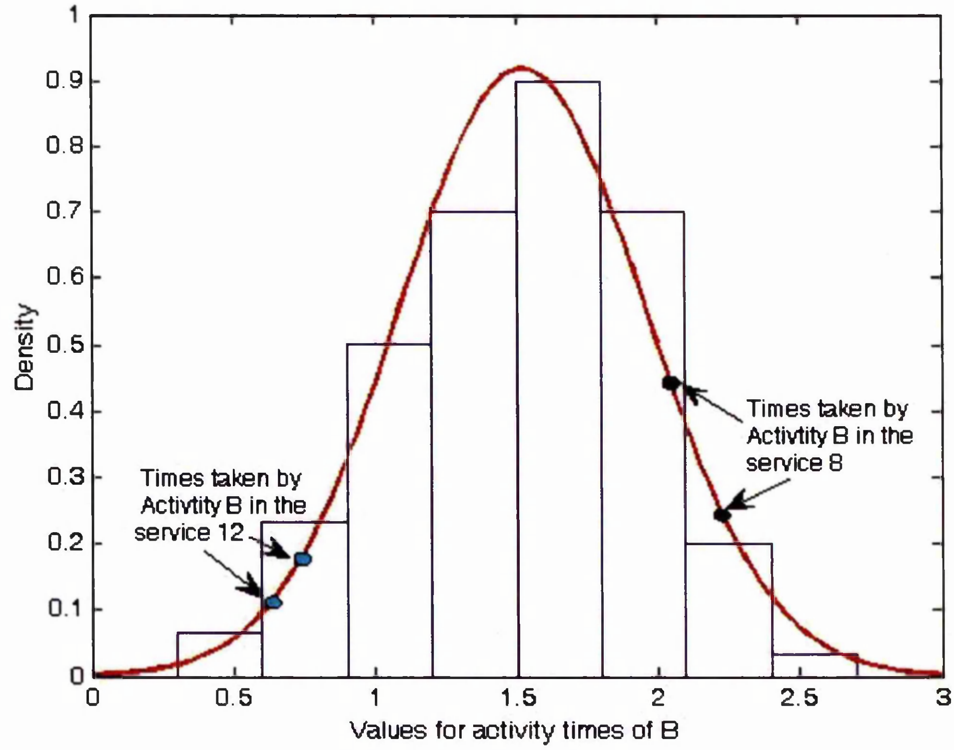


Figure 7.13: Values for Activity Times of B

Figure 7.13 shows that in service 8, repeating caused the total time taken by activity B to be about 4.5 hours which, combined with the time taken by other activities, caused the total time taken by the system to exceed 10 hours; that is, a system failure. In service 12, repeating caused the total time taken by activity B to be about 1.5 hours only. However, the repetition of activity D did not significantly increase the total time taken, and the accumulated time taken by the system was less than 10 hours.

Analogously, failures of the same activities can lead to different results of system performance. For example, in both service 1 and 7, activity A is repeated once, but service 1 was completed on time while service 7 failed.

The results also indicate that the failure of activity C can hardly cause the system to fail unless combined with failure(s) of other activities. In service 3, even though the activity failed 3 times, the system could still complete on time.

Failures / Services	Activity A	Activity B	Activity C	Activity D	System Failure
Service 1	1	0	0	0	0
Service 2	0	0	0	1	1
Service 3	0	0	3	0	0
Service 4	0	1	0	0	1
Service 5	1	0	0	1	1
Service 6	0	1	1	0	1
Service 7	1	0	0	0	1
Service 8	0	1	0	0	1
Service 9	0	0	0	3	1
Service 10	0	0	1	1	1
Service 11	1	1	0	0	1
Service 12	0	1	0	1	0

Table 7.7: Number of failures of each activity and system failures

The fact that no explicit causality exists leads to a consideration of how to identify the critical activities that have the greatest influence on system reliability. To do so can provide important instructions for system resource re-allocation, or giving more resource to the critical activities.

7.6.4 Sensitivity Analysis

An important use of the design support tool is to find the critical activities that are threatening system reliability. Simulations can be run using different resources (and hence cost) and different reliabilities for each activity to explore the sensitivity of the system to the performance of particular activities. Thus, system activities may be resourced to give, as near as possible, the optimized system performance.

7.6.4.1 Sensitivity Analysis with Respect to Resource for Activities

The sensitivity of the system reliability to the resources allocated to each section can be explored. Simulations are run with changing resources given to each activity.

Simulation results show that the system reliability shows an obvious improvement as the resources for activity A increase; see figure 7.14. However, the influence of the increasing resource becomes inconspicuous from $\boldsymbol{C_a}$ = 2.3. Thus, it is suggested that the resource given to activity A should be not more than 2.3 persons.

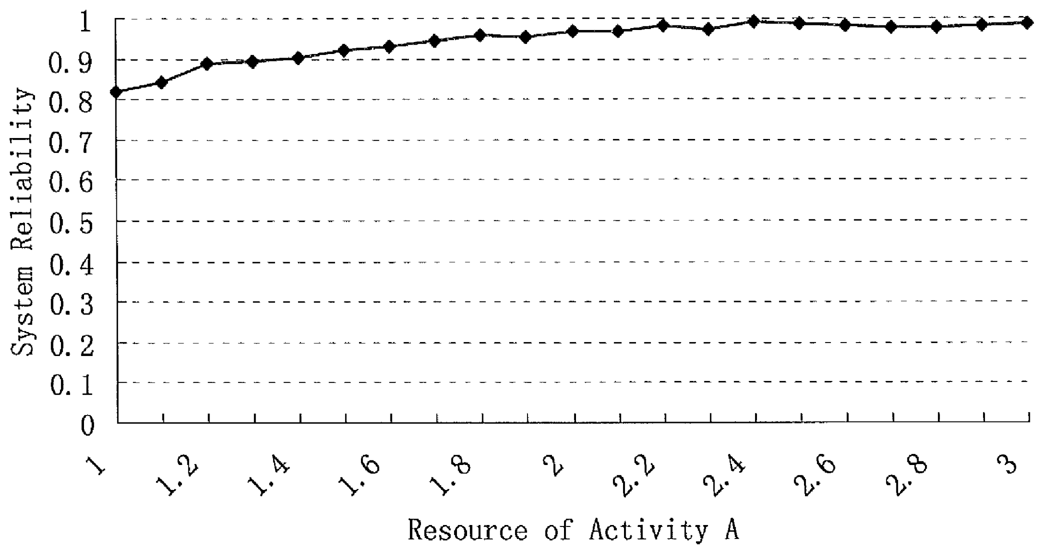


Figure 7.14: System Reliability against Resource of Activity A

For activity B, increasing its resources did not result in as much system reliability improvement as it did in activity A. Such resource increase hardly brings the system reliability above 0.9; see figure 7.15.

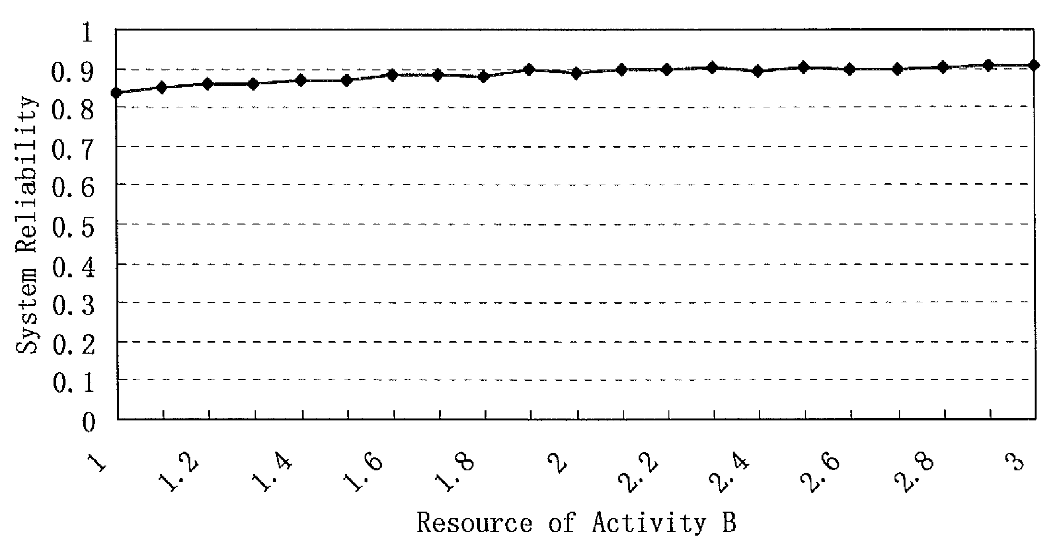


Figure 7.15: System Reliability against Resource of Activity B

The simulation results indicate that giving more resources to activity C would not cause the system to be more reliable; see figure 7.16. As the time consumed by the activity has already been very short, that is 0.5 hour, increasing its resources is unlikely to reduce it further. Another reason is that the activity is carried out almost simultaneously with activity B, which normally consumes much more time; therefore, the time consumed by activity C is shadowed when calculating the total service time. Thus, it can be concluded that the increase in resource given to activity C would not make a significant contribution to the system reliability improvement.

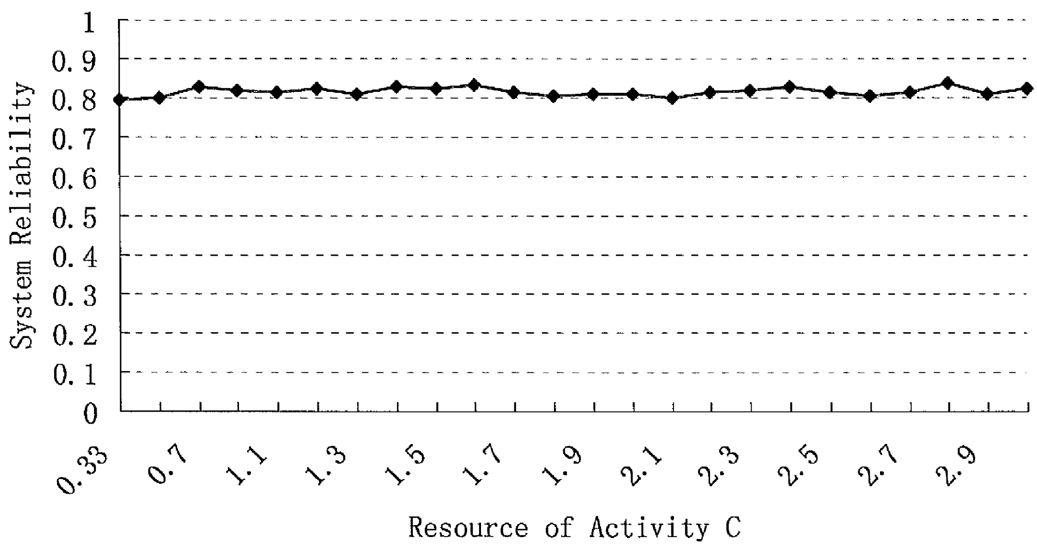


Figure 7.16: System Reliability against Resource of Activity C

On the other hand, the resource allocation of activity D has an obvious influence on the reliability of the whole system. However, as with activity B, the system reliability increases from about 0.83 and stops at about 0.9 when the given resource is 1.8 persons; see figure 7.17.

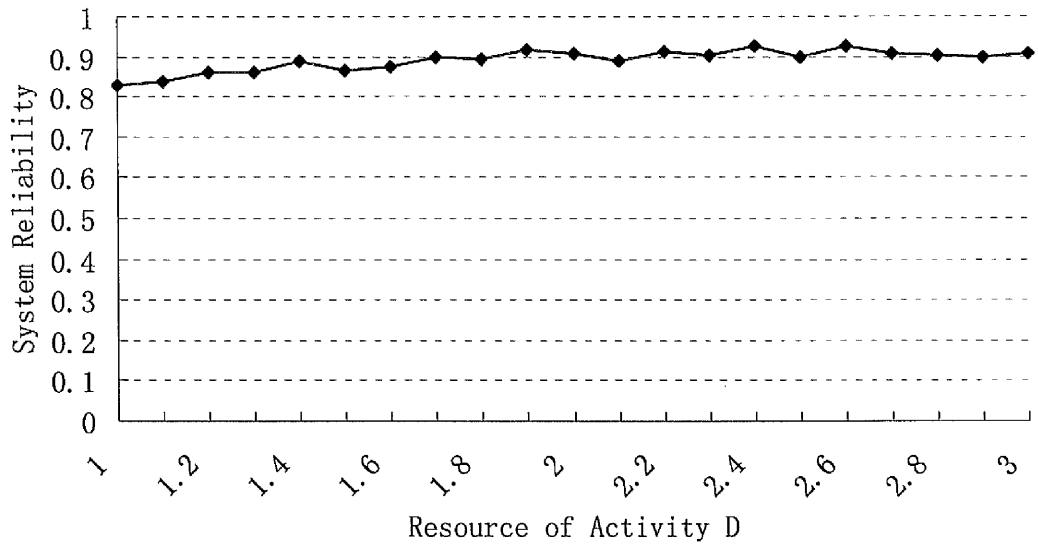


Figure 7.17: System Reliability against Resource of Activity D

Among all the activities, activity A has the greatest impact on system reliability. When its resource equals 2.3 persons, system reliability increases to over 0.95. Evidently, system reliability is more sensitive to the resource of activity A. Thus, it is can be concluded that the most effective way to improve system reliability is reducing the time spent on activity A.

7.6.4.2 Sensitivity Analysis with Respect to Reliability of Activities

The sensitivity of the system to the reliability of individual activities can readily be explored. The reliability of each activity was varied from 0.5 to 1.0 and the effects on the system reliability were calculated. Figure 7.18 shows the results for activities A, B, C and D respectively.

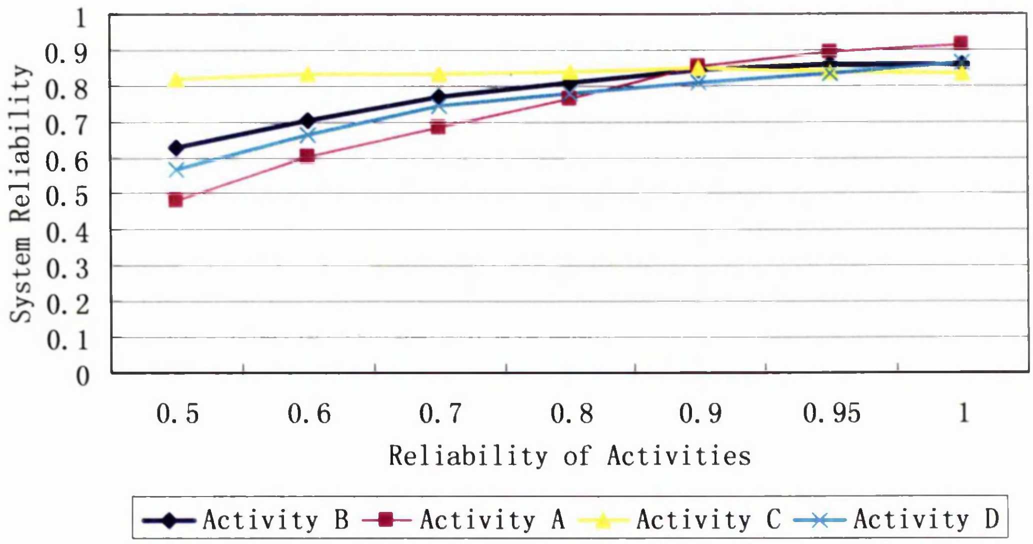


Figure 7.18: System Reliability against Reliability of Activities

Activity A shows the greatest influence on system reliability, which improves from below 0.5 to above 0.9 as the reliability of activity A increases.

Comparatively, activities B and D reveal less influence on the whole system.

The reliability of the system does not obviously improve with an increase in reliability of activity C, which again reveals that this activity is not critical with respect to system reliability.

7.6.5 Improving the Use of Resources

Sensitivity analysis indicates that the resource configurations of activities A and D are critical in implementing the contracted services successfully; therefore, it is possible to improve system performance by giving more resources to them. Moreover, resources given to activity C could be reduced. A simulation was run given the adjusted values displayed in table 7.8.

Parameters / Activity	Resources (person) (C_a)	Activity Time (hour)		Reliability (R_a)
		Mean (T_m)	Deviation (T_d)	
Activity A	1.8	2.2	1	0.9
Activity B	1	1.5	0.5	0.9
Activity C	1/3	1.5	1/6	0.9
Activity D	1.5	1.33	1/3	0.9

Table 7.8: Increased Resources of the SSS

The results show that increasing the total resources to 4.633 units can cause system reliability to achieve 0.98; see figure 7.19.

Moreover, the system performance could be improved using reduced resources. For example, as shown in the table 7.9, system resources are reduced to 3.83; moving some resource from activity C to activity A, system reliability can still achieve 0.91; see figure 7.20.

Parameters / Activity	Resources (person) (C_a)	Activity Time (hour)		Reliability (R_a)
		Mean (T_m)	Deviation (T_d)	
Activity A	1.5	2.67	1	0.9
Activity B	1	1.5	0.5	0.9
Activity C	1/3	1.5	1/6	0.9
Activity D	1	2	1/3	0.9

Table 7.9: Reduced Resources of the SSS

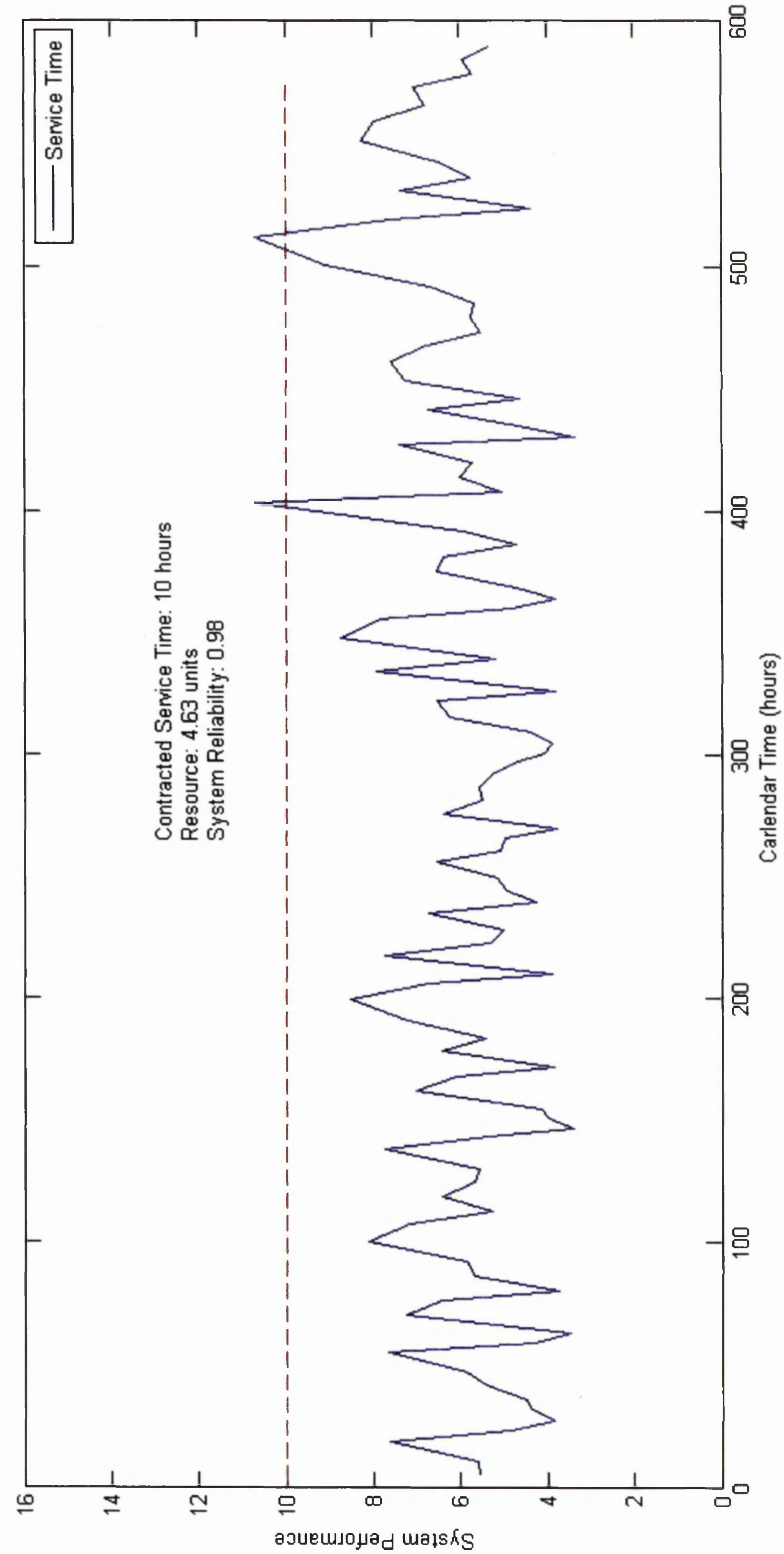


Figure 7.19: Improved System Performance with Increased Resources

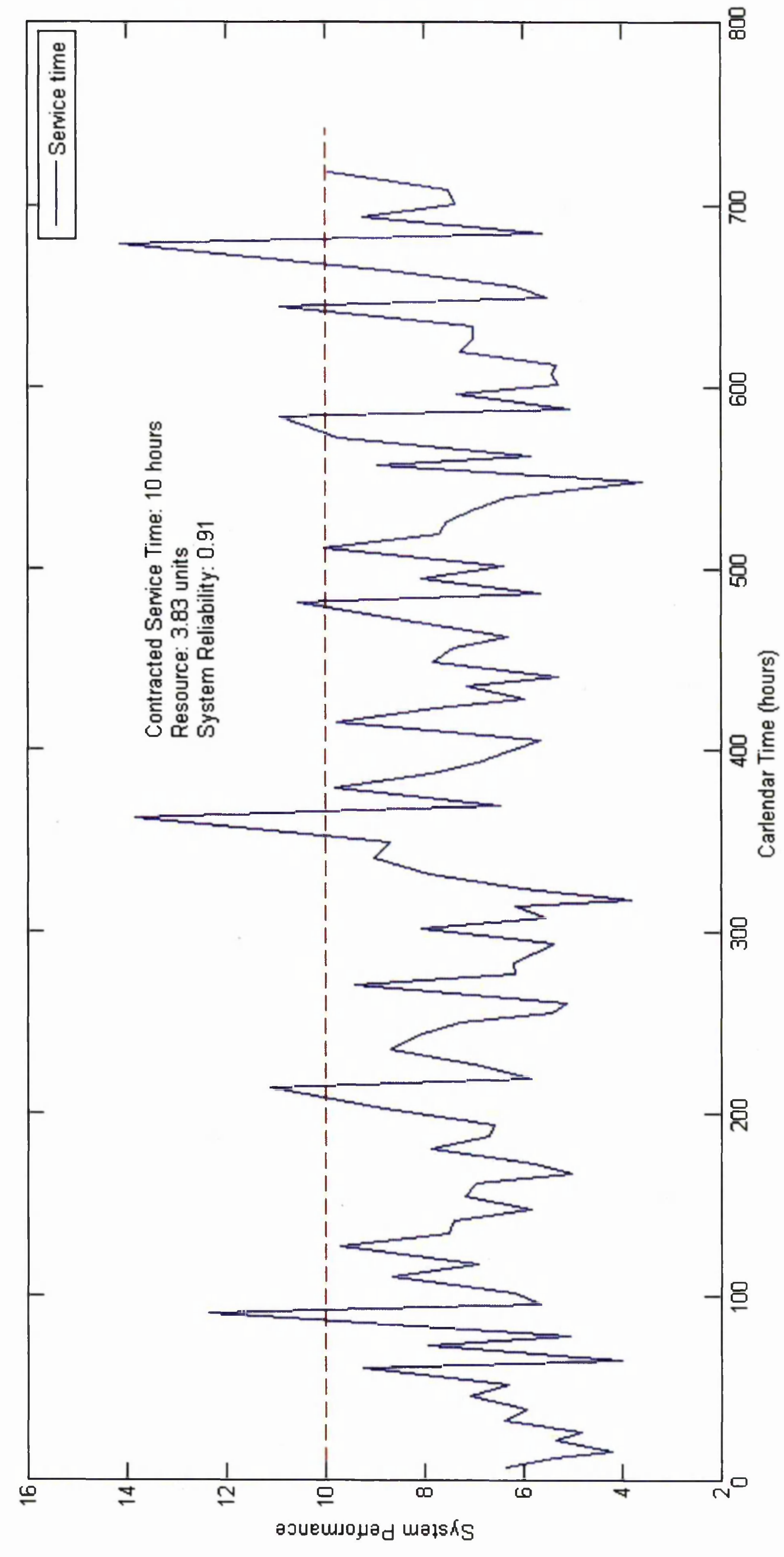


Figure 7.20: Improved System Performance with Reduced Resources

7.7 Conclusion

Contract and customer-perceived failures are important. Contract failures are when the system performance with respect to defined KPIs falls below specified levels. Customer-perceived failure can occur if performance falls below expectations even though contracted performance has been achieved. For example, if a high level of performance has been routinely given then that may well become the reference rather than the contracted KPI.

The functional reliability of a SSS may be measured by monitoring its KPIs, for example the service time. System modelling is a useful design support tool and it also enables performance evaluation of the developed service system. Each activity may be simply modelled using the time taken for the activity, the resources used each time the activity is carried out, and the reliability of the activity.

The design support tool can model the system performance and highlight internal and external failures. Sensitivity analysis can be carried out so as to find critical activities with respect to system reliability. Thus, the performance of SSSs can be improved by reallocation of resources among activities.

Chapter Eight

Case Study

8.1 Introduction

This chapter presents a case study that applies the proposed 'top down' service design approach to an industrial problem. The case is concerned with service operations that aim to achieve the contracted availability of total supported aircraft engines that are supplied by the company B (see chapter 6). The case study involves applying the 'top down' service design approach to obtain a comprehensive SSS and uses the system modelling method to evaluate its performance.

This chapter begins by explaining the background to the total support of aircraft engines provided by the company and the strategy of the maintenance, repair and overhaul. The 'top down' design approach is then employed to create the system diagrams of the supporting services. The performance of the SSS thus obtained is then evaluated with respect to reliability using the system modelling method. The sensitivity of the system reliability to particular activities is investigated to find critical activities, which provides references for improving the use of resources and the system performance. The chapter ends by suggesting the improved allocation of resources.

8.2 Functional Aircraft Engine

8.2.1 Strategy of Service Operations

The company offers functional components for aircraft and rocket engines and provides services for jet engines for military and civil aircrafts. Currently it has more than 25 Airbus 340-500 aircrafts on a Power Contract (PC) agreement, that is, a functional engine provision against an agreed rate per flying hour fee. It has been agreed that the aircraft of clients shall never stay on the ground for more then 24 hours, which means that the company has to change or repair the engine within 24 hours.

The company endeavors to establish an effective and reliable SSS to assure the function provision to meet the contracted level. Since aircraft engines are safety-critical products, any problem that occurs needs to be inspected comprehensively. In some cases, it is not possible to repair an engine on the wing and remanufacture is required. Such overhauls take weeks to carry out. Therefore, the policy of the company is to achieve the promised availability by implementing the service on an engine change basis. Thus, the recovered engine is remanufactured and stored in the service base for future change.

8.2.2 Service Blueprint

The figure 8.1 shows the blueprint of the prospective service operations which contain a process for delivering a functional engine to a client. Other functions include the process of malfunctioning engine remanufacture and storage for future use, purchase of spare parts from external suppliers and further analysis of unacquainted engine failures which is referred as handle deviations.

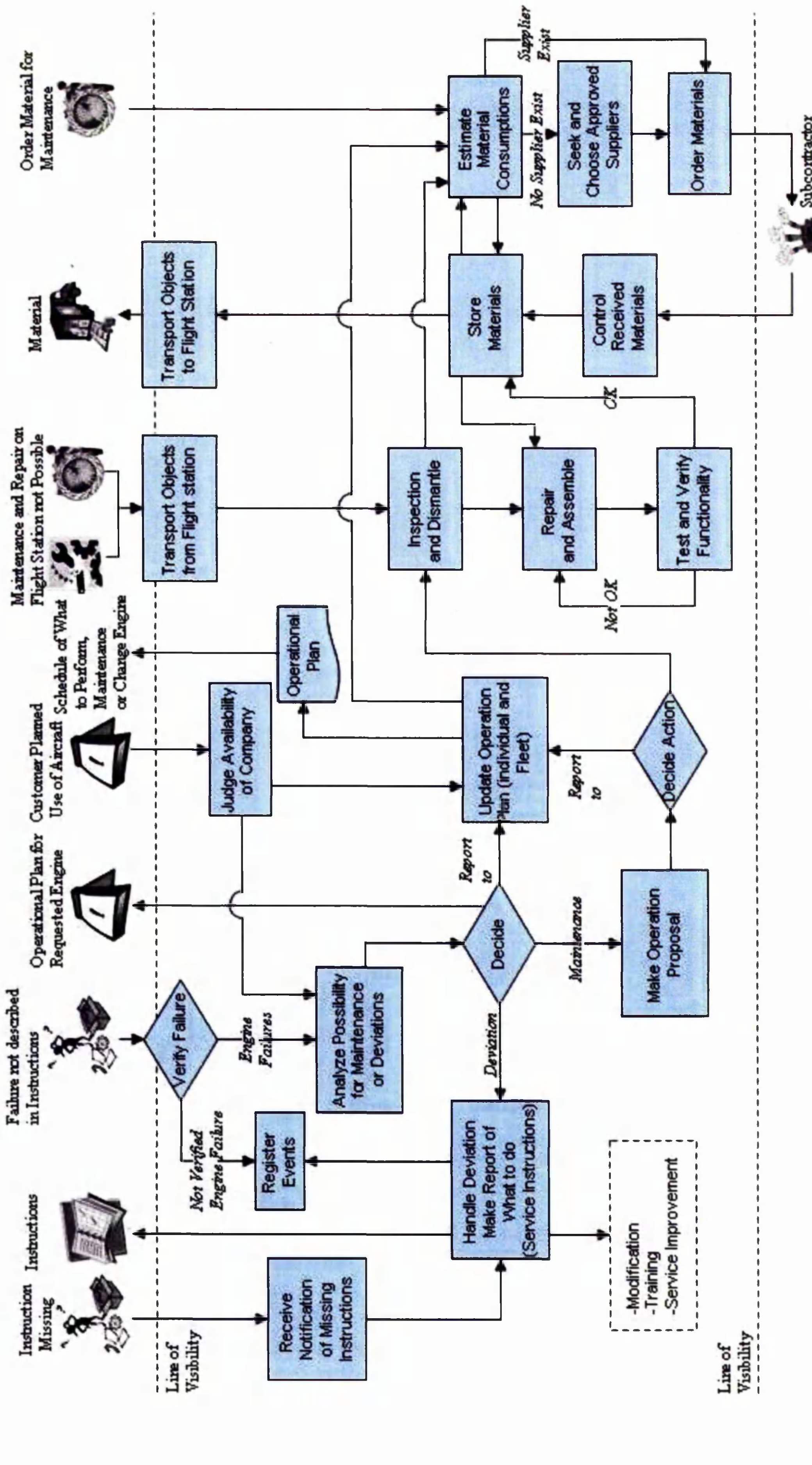


Figure 8.1: Adapted Blueprint for Supporting Services of the Company (Persson, 2004)

8.3 System Diagrams

8.3.1 Equivalent Activities in the GSSS

Table 8.1 shows a comparison between the activities of the service operations of the company and their counterparts in the GSSS.

Activities of the Company B's SSS	Equivalent Activities in the GSSS
Receive notification of missing instructions	Collect information from client
Verify failure	Collect information from client
Register events	Database: Request analysis
Judge availability of company	Analyse request
Analyse possibility for maintenance or deviations	Analyse request
Decide	Analyse request
Handle deviation, make report of what to do	Database: Request analysis
Make operation proposal	Analyse information
Decide action	Make decision
Update operation plan	Make decision
Transport objects from flight station	Transport objects from client's site
Inspection and dismantle	Arrival inspection
Repair and assemble	Maintenance/repair hardware
Test and verify functionality	Test hardware
Store materials	Hardware storage
Estimate material consumptions	Check stock level
Seek and choose approved suppliers	Order hardware
Order materials	Order hardware
Control received materials	Receive hardware
Transport objects to flight station	Transport objects to client's site

Table 8.1: Equivalent Activities of the Company in the GSSS

The activities 'Judge availability of company', 'Analyse possibility for maintenance or deviations' and 'Decide' are all parts of the GSSS activity 'Analyse request'. In this case, the function 'Analyse request' is therefore required to be designed into a further detailed level; thus, it becomes a sub-system comprising its three parts. Similarly, the GSSS elements 'Collect information from client', 'Make decision' and 'Order hardware' are broken down into further activities (see section 8.3.2).

8.3.2 System Diagrams

8.3.2.1 Creation of the System

Thus, the system diagrams were obtained as follows:

Step 1: Activities in the GSSS that are not included in the service blueprint were excluded;

Step 2: Sets of activities from the service blueprint that combine to form a single activity in the GSSS were combined to form a new sub-system of the SSS. Thus, the original single activity in the GSSS is replaced by a new sub-system with the same function;

Step 3: Remove unneeded input(s) and output(s) and make a compatibility check on the remaining elements;

Figure 8.2 shows the top-level description of obtained system diagrams. On the highest level of the system, five main functions are kept. The details of each main function are explained in the following sections.

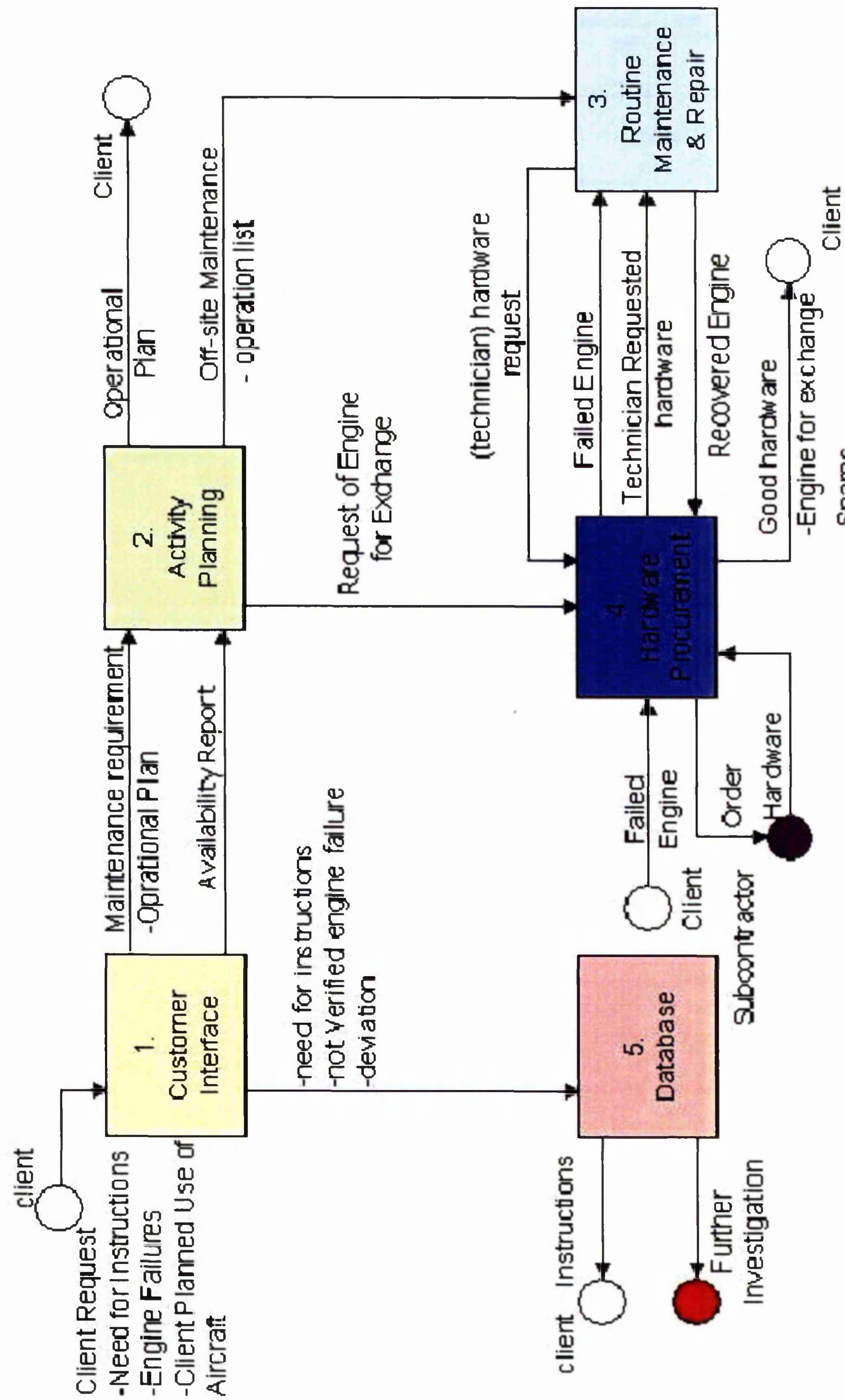


Figure 8.2: Top-Level Description of the Obtained Service Support System

8.3.2.2 Customer Interface

The customer interface collects and analyses requests from clients. In this system, client requests are notification of engine failures and need for instructions; see figure 8.3. Two activities are implemented in 'Collect and Sort Information from Client' and they work independently; see figure 8.4. 'Receive Notification of Missing Instruction' handles the requests for instructions and send the request directly to the 'Database' where the instructions are printed and delivered. 'Verify Failure' is implemented should a failure happens to an engine. After diagnosing the problem, the information of the confirmed failure is sent for further analysis. Should the failure is unknown, the information will be sent to the 'Database' to register the event and arrange for further investigation. In addition, another input from the client is the client's planned use of hardware.

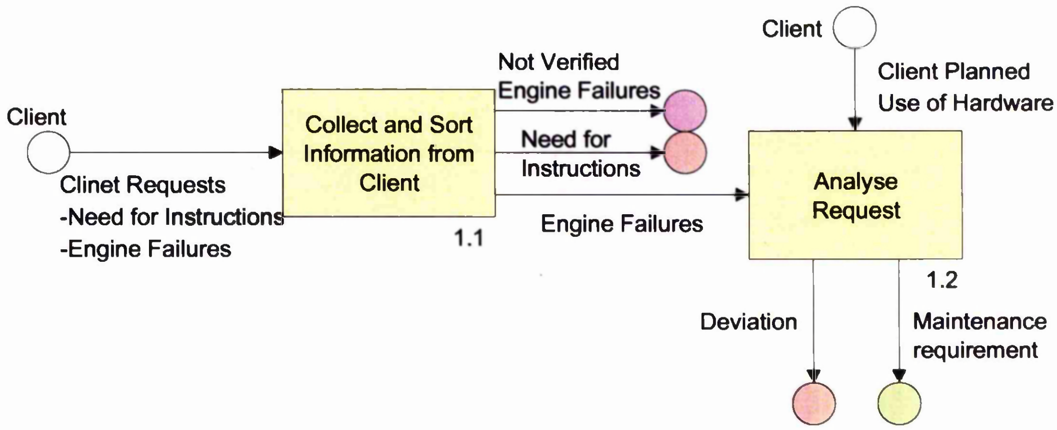


Figure 8.3: Second Level of Customer Interface

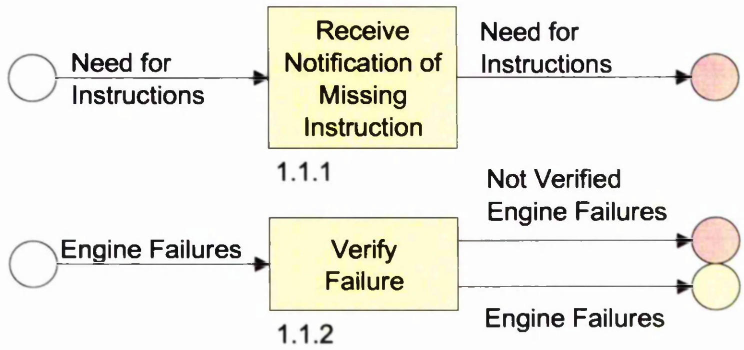


Figure: 8.4: Third level of Customer Interface: Collect and Sort Information from Client

The information of the confirmed engine failure is then sent to the sub-system 'Analyse Request' where the possibility for maintenance or deviation is analysed. In the sub-system, three activities 'Judge Availability of Company', 'Analyse Possibility for Maintenance or Deviation' and 'Decide Maintenance or Deviation' are carried out; see figure 8.5. Given two inputs: client planned use of aircraft and information of engine failure; these three activities generate an availability report and the decision to implement maintenance or deviation.

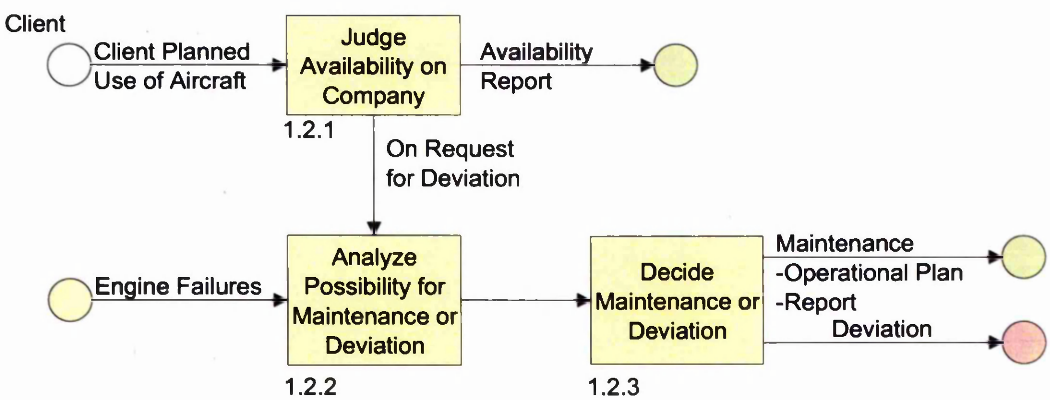


Figure: 8.5: Third level of Customer Interface: Analyse Request

In order to undertake the analysis, reference to the planned engine usage of the operators is required. Therefore, an availability report is produced. The report is also sent to the 'Activity Planning' function for updating service operation plan.

If it is decided to proceed with maintenance, an operational plan and report that records failure diagnosis information will be sent to the 'Activity Planning' function as well, otherwise the service goes to an activity that authorises deviations.

8.3.2.3 Activity Planning

The second level of 'Activity Planning' contains only a sub-system of 'making decision' which creates an operational plan, decides actions concerning off-site maintenance and sends the request for hardware exchange; see figure 8.6.

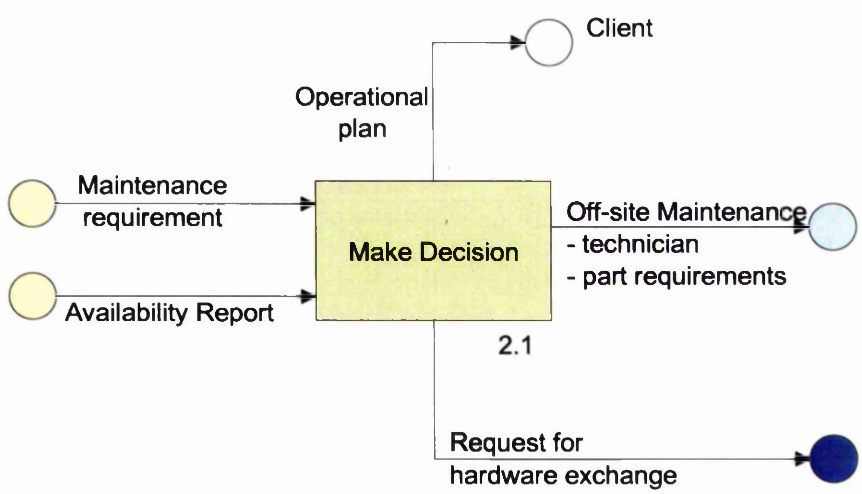


Figure 8.6: Second Level of Activity Planning

The decisions are made by three steps which are 'Make Operation Proposal', 'Decide Actions' and 'Update Operational Plan'. Given the inputs 'Maintenance Requirement' and 'Availability Report', an operation proposal is made firstly and then sent for deciding actions concerning off-site maintenance. Drawing on the proposal, the actions decided and the availability report, the final operation plan is then updated and delivered to the client. Meanwhile, a request for engine exchange is sent out; see figure 8.7.

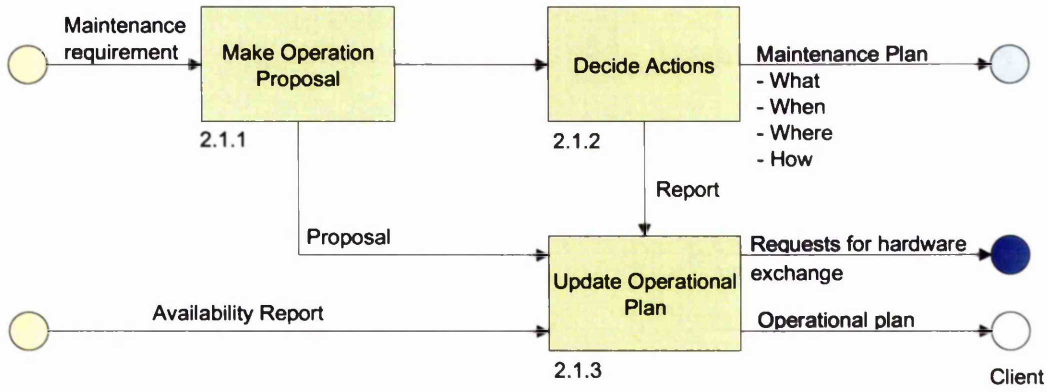


Figure 8.7: Third Level of Activity Planning

From the activity planning, the service follows two parallel routes. One route is carrying out the off-site maintenance; that is, restoring the function of the malfunctioning engine in workshops. Meanwhile, a request for an identical engine is sent to the 'Hardware Procurement' function that is responsible for delivering it to the waiting client.

8.3.2.4 Routine Maintenance and Repair

Since the repair of malfunctioning engine takes much longer than change the engine, restoring the engine function is designed to be off-site maintenance; see figure 8.8.

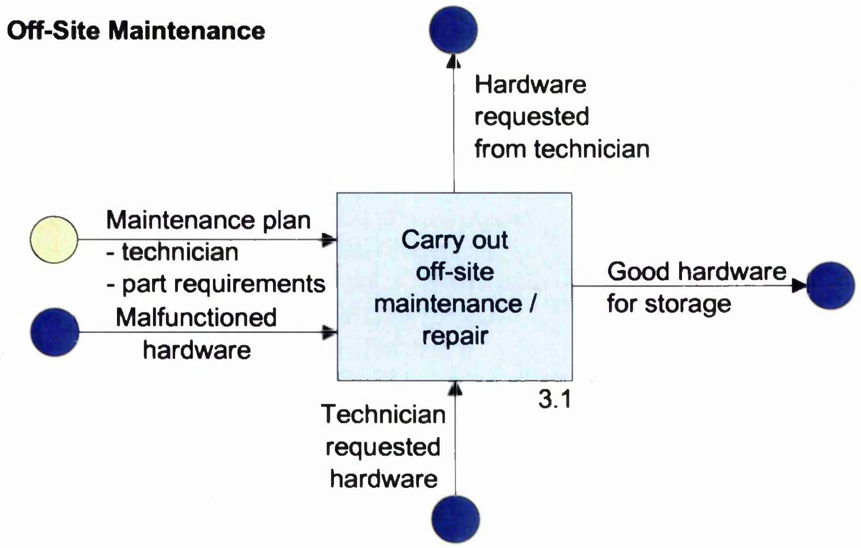


Figure 8.8: Second level of Routine Maintenance and Repair

When the malfunctioning engine and the plan of actions on hardware are ready, three activities to recover its function start working; see figure 8.9. The engine is first dismantled and inspected while deciding if spare parts are required. Then the actions on the hardware are implemented, which is followed by a functionality test. If the engine functions well, it will be stored and for future use.

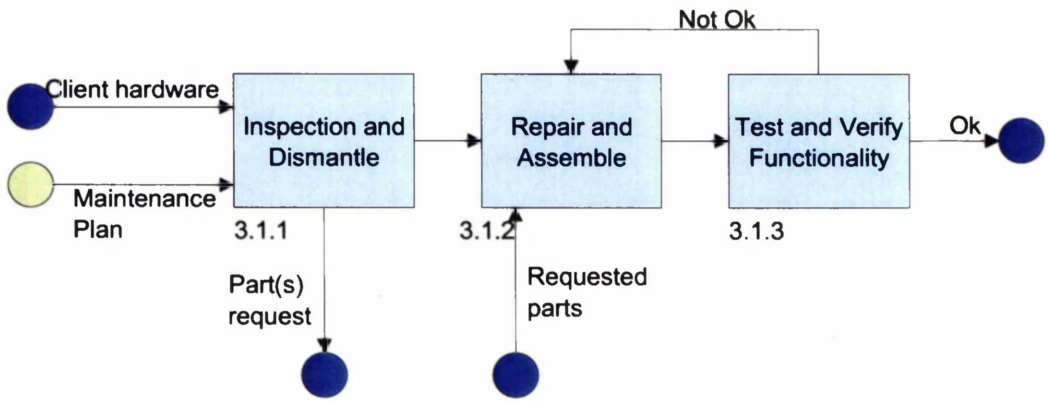


Figure 8.9: Third Level of Routine Maintenance and Repair

8.3.2.5 Hardware Procurement

'Hardware Procurement' is responsible for transporting malfunctioning engines to the workshop and good engines to clients. The spare parts for use in the workshop or by technicians are provided if required. In addition to the function of transportation, this sub-system involves activities dealing with the purchase of spare parts from external suppliers should the requested ones be out of stock; see figure 8.10, 8.11 and 8.12.

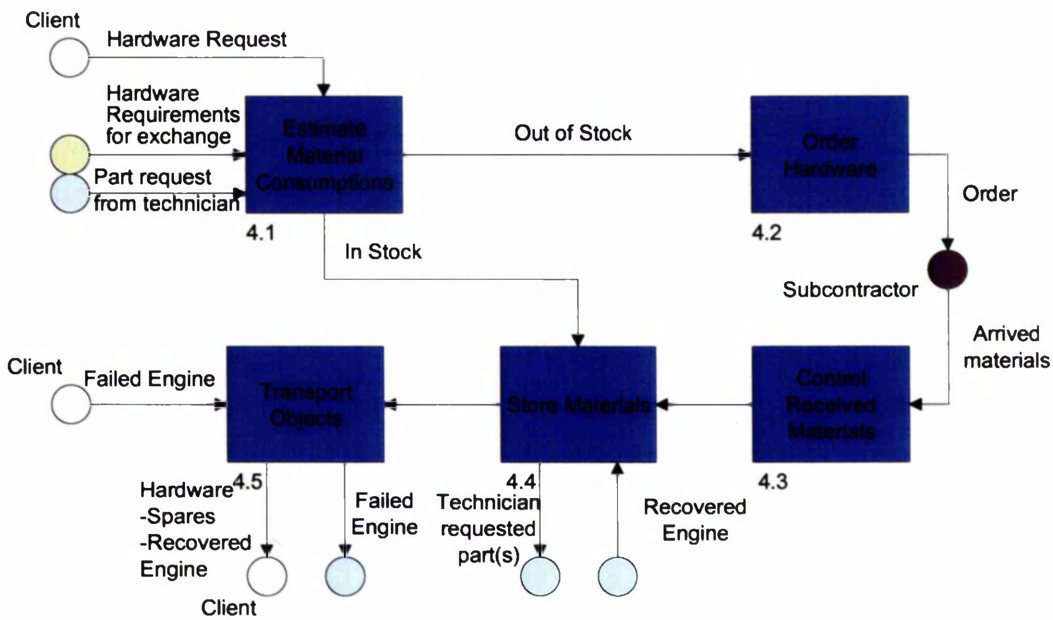


Figure 8.10: Second Level of Hardware Procurement

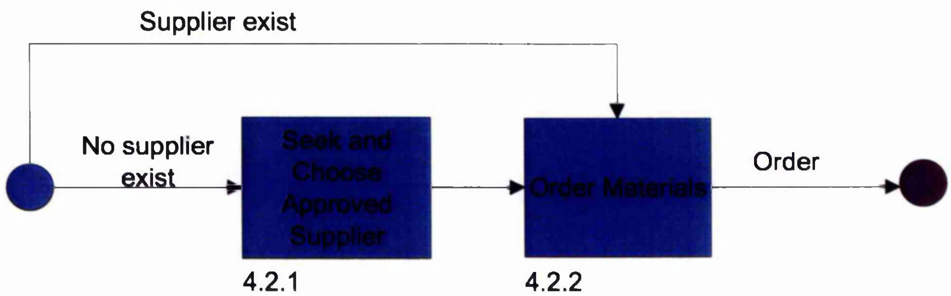


Figure 8.11: Third Level of Hardware Procurement: Order Hardware

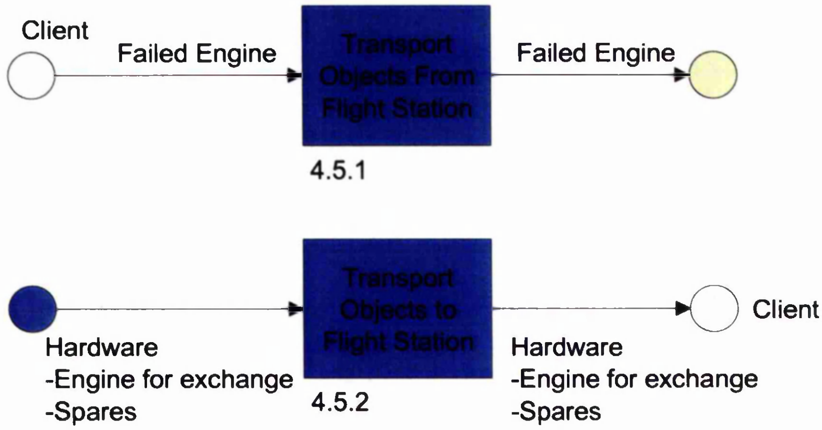


Figure 8.12: Third Level of Hardware Procurement: Transport Objects

8.3.2.6 Database

In this function, information from the client is considered and instructions to client are generated; see figure 8.13. Also the need for further investigation is identified should further investigation of a hardware failure be required.

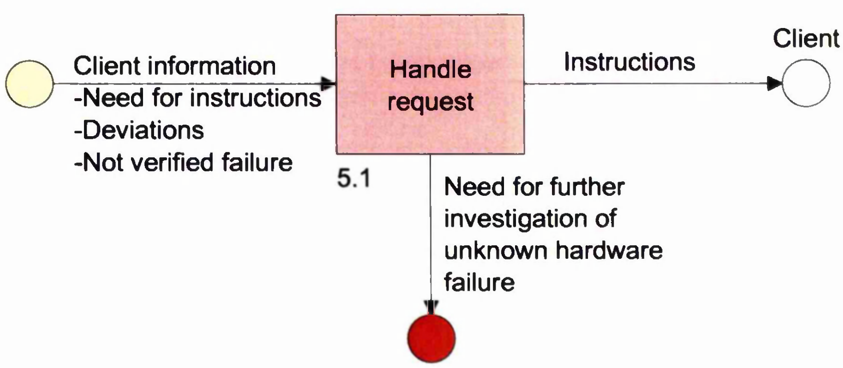


Figure 8.13: Second Level of Database

In this sub-system, two separated activities are designed: 'handle deviation' and 'register events'; see figure 8.14. Finding and sending client requested instructions are implemented in 'handle deviation'.

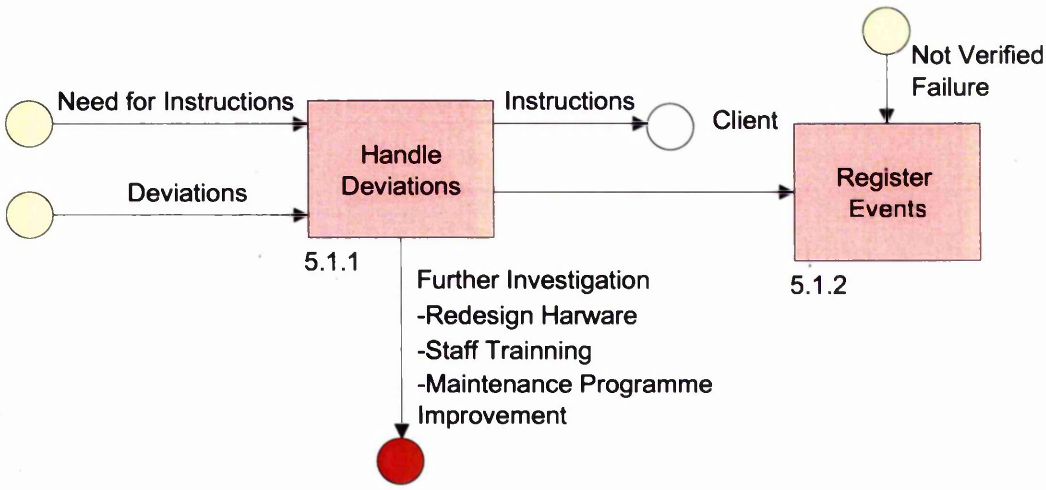


Figure 8.14: Second Level of Database

8.3.3 Complete System

Sections 8.3.2.1 to 8.3.2.6 have described how the system components were derived using the 'top down' design method. Figure 8.15 shows the complete system with all activity inputs and outputs. The system in figure 8.15 was then modelled and simulated in the reliability assessment; see section 8.4.

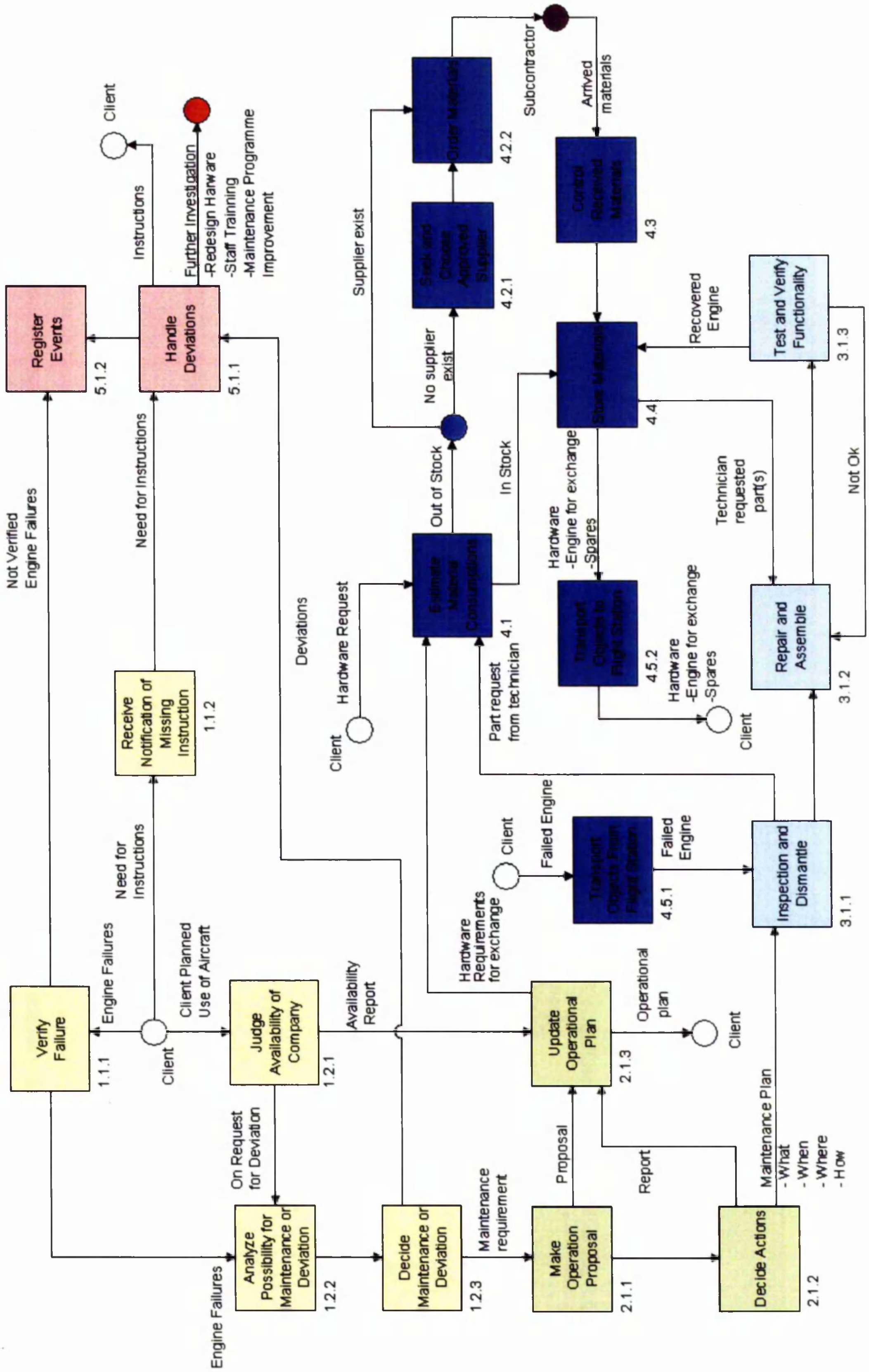


Figure 8.15: Complete Service Support System

8.4 Reliability Assessment of the SSS

8.4.1 System Reliability: Criticality Assessment

The objective is to identify those activities that have the greatest effect on system reliability and to determine the optimum use of resources in the system.

An initial resource allocation, with assumed reliabilities was made as shown in the table 8.2. The system model was then run 100 times and the result is shown in figure 8.16. There are several instances of long service time, but most are below 24 hours. If the assumed contract time is 24 hours, then the system reliability is 0.71 (24 hours is a typical value).

The approach taken to criticality assessment is as follows. First, the resource allocated to each activity is altered and the effect on the system service time is noted with respect to the reference system reliability of 0.71. Then, the effect of variation in the reliabilities of individual activities is similarly explored. Finally, the resources are adjusted for each activity to give an overall improvement in the system reliability but with no overall increase in resource.

Activity \ Parameters	Resources (person) (C_a)	Activity Time (hour)		Reliability (R_a)
		Mean (T_m)	Deviation (T_d)	
Receive notification of missing instructions	1	1/6	1/20	0.9
Verify Engine Failure	1	4	1	0.9
Register events	1	1/6	1/20	0.9
Judge Availability	1	0.5	0.1	0.9
Analyse Possibility for Maintenance or Deviation	1	1.5	0.5	0.9
Decide Maintenance or Deviation	1	0.5	1/6	0.9
Handle Deviation	1	0.5	1/6	0.9
Make Operation Proposal	1	1	1/3	0.9
Decide Actions	1	1.5	1/6	0.9
Update Operational Plan	1	1	0.2	0.9
Transport objects from flight station	1	7.5	1.5	0.9
Inspection and dismantle	1	2	1/3	0.9
Repair and assemble	1	5	1.5	0.9
Test and verify functionality	1	1	0.2	0.9
Store materials	1	2	1/3	0.9
Estimate Material Consumption	1	1.5	0.2	0.9
Seek and choose approved suppliers	1	1	0.2	0.9
Order materials	1	1.5	0.5	0.9
Control Received Materials	1	2	1/3	0.9
Transport to Flight Station	1	7.5	1.5	0.9

Table 8.2: Values of Parameters of the SSS

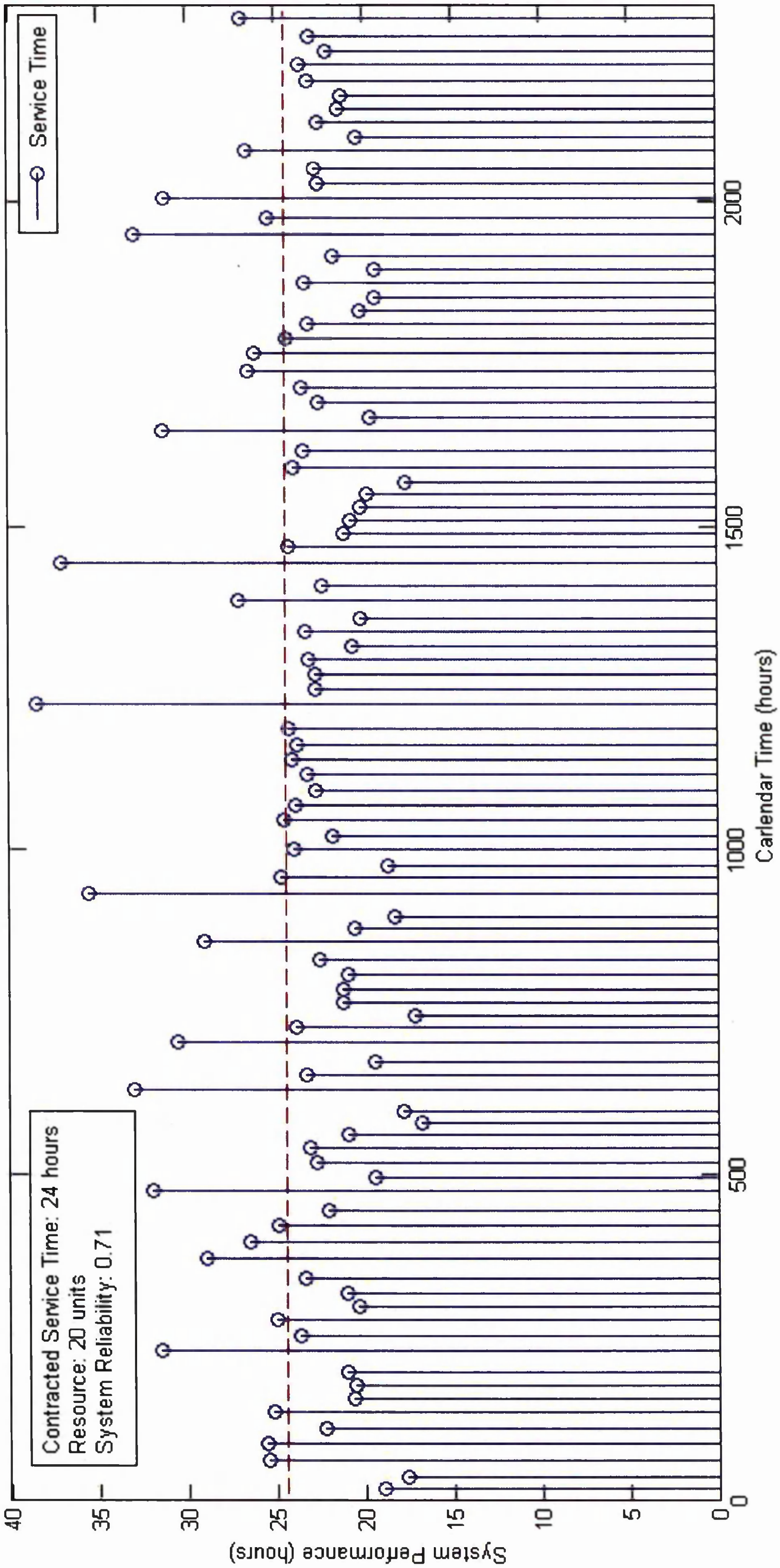


Figure 8.16: Risky System Performance

8.4.2 Sensitivity Analysis with respect to Resource Given to Activities

Critical activities can be revealed by adding resource to each activity in turn and running the simulation. Table 8.3 shows the increase in system reliability that is obtained by increasing the resource of each activity by 100%, leaving all other resources at their initial values for each simulation.

Activities of the Company B's SSS	**Increase of Resource (%)**	**Increase of System Reliability (%)**
Receive notification of missing instructions		
Verify failure	100%	21%
Register events	100%	0.4%
Judge availability on company	100%	5.4%
Analyse possibility for maintenance or deviations	100%	11.7%
Decide maintenance or deviation	100%	4.1%
Handle deviation (make report of what to do)	100%	1.3%
Make operation proposal	100%	10.4%
Decide actions	100%	8.2%
Update operation plan	100%	6.3%
Transport objects from flight station	100%	2%
Inspection and dismantle	100%	0.9%
Repair and assemble	100%	-0.07%
Test and verify functionality	100%	2.4%
Store materials	100%	10.6%
Estimate material consumptions	100%	10.3%
Seek and choose approved suppliers		
Order materials		
Control received materials		
Transport objects to flight station	100%	32%

Table 8.3: Reliability improvement of the SSS

Under normal service operations, the activities 'Receive notification of missing instructions', 'Seek and choose approved suppliers', 'Order materials', 'Control received materials' are not carried out. Therefore, they were excluded from this criticality investigation. Among other activities, some show considerable influence on the system performance, for example 'Verify failure' and 'Transport objects to flight station'; some show obvious influence, for example 'Update operation plan'.

Several activities seem having little help to improve the system performance. This is because these activities are not involved in the critical route to deliver a functional engine to clients. Among them, 'Register events' and 'Handle deviation' are carried out in the parallel manner with the critical route. 'Transport objects from flight station', 'Inspection and dismantle', 'Repair and assemble' and 'Test and verify functionality' form the process of implementing the off-site maintenance which could take weeks and work in the parallel manner with the critical route as well. The negative increase of the system performance aroused by the activity 'Repair and assemble' is due to the uncertainty of the system performance.

Further sensitivity analysis is carried out to the activities that show obvious relevance with the system performance. Such a simulation can be run using different resources (and hence cost) and different reliabilities for each activity to explore the sensitivity of the system to the performance of particular activities. Thus, system activities may thus be resourced to give, as near as possible, the optimized system performance.

Simulations are run with changing resources given to each activity within the range 1.1 to 3 units. Figure 8.17 shows that the system reliability has considerably improve as the resource of 'Verify Engine Failure' increases. However, the influence of the increasing resource becomes inconspicuous from $\boldsymbol{C_a}$ = 2.4 units. Thus, it is suggested that the resource given to 'Verify Engine Failure' should not be more than 2.4 persons.

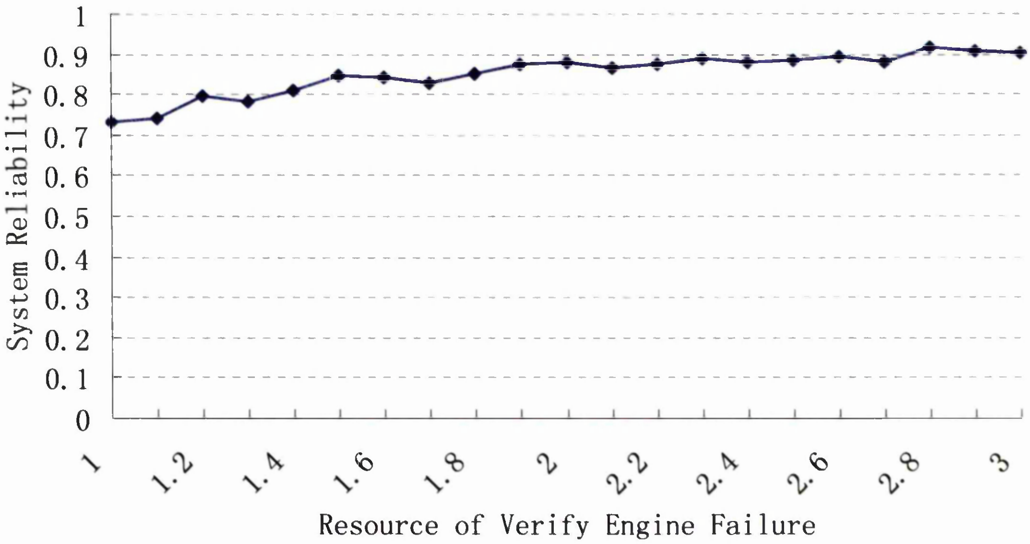


Figure 8.17: System Reliability against Resource of Verify Engine Failure

The activity 'Judge Availability' seems having no any positive influence on system reliability; see figure 8.18. The explanation is that the activity is carried out almost simultaneously with 'Verify Engine Failure' which normally consumes much more time. Thus, it can be concluded that increase the resource given to the activity would not have considerable contribution to the system reliability improvement. Alternatively, it is possible to reduce the resource given to the activity to a certain level.

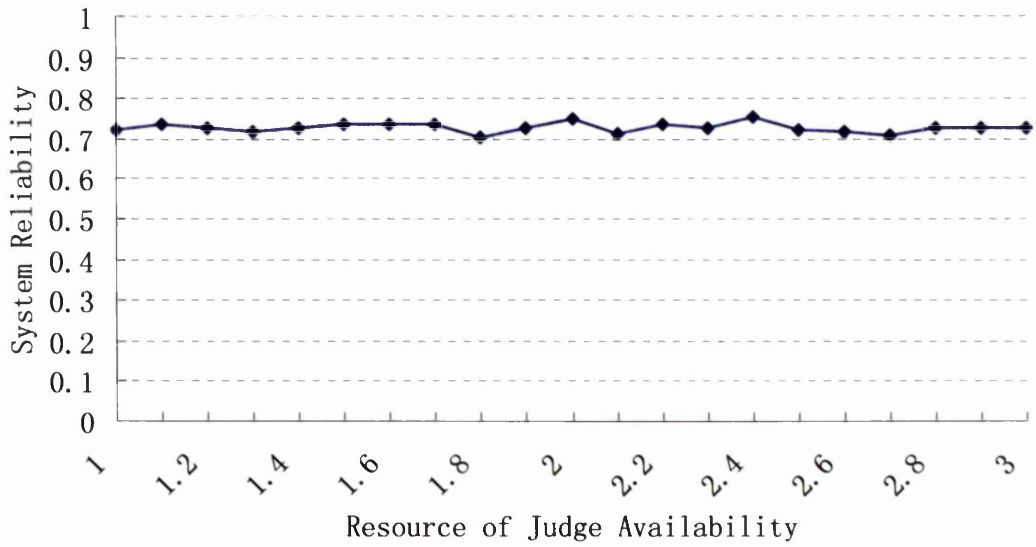


Figure 8.18: System Reliability against Resource of Judge Availability

Comparatively, the resource of activity 'Analyse Possibility for Maintenance or Deviation' showed greater influence on the system reliability; see figure 8.19. The system reliability increases from about 0.71 and stops at about 0.8 when the given resource is C_a = 2.1 units.

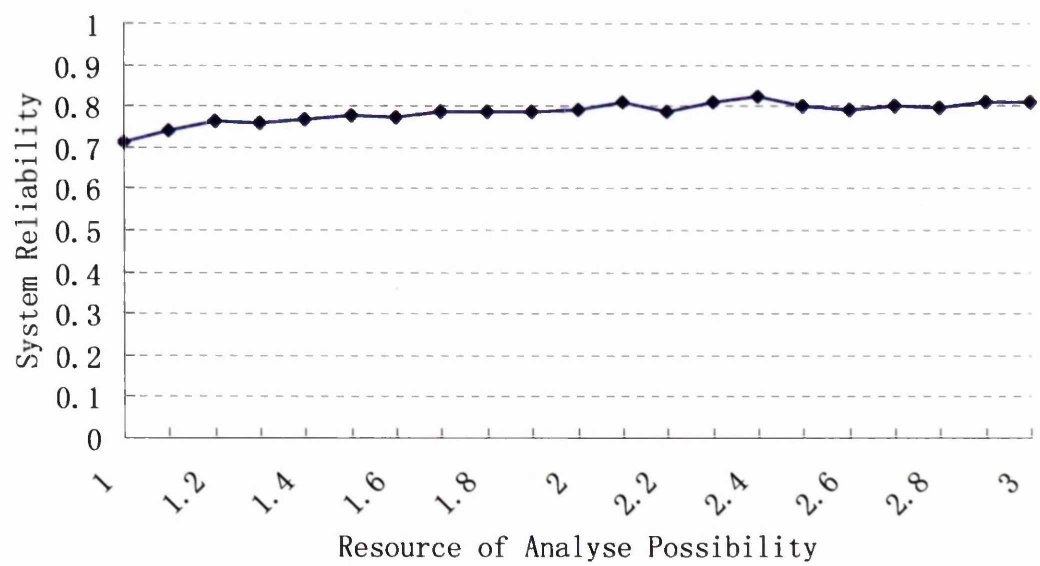


Figure 8.19: System Reliability against Resource of Analyse Possibility for Maintenance or Deviation

Nevertheless, not all activities have obvious contribution to the system reliability improvement. For example, simulation results indicate that giving more resources to 'Decide Maintenance or Deviation' would not cause the system to be more reliable; see figure 8.20. As the time consumed by the activity has already been very short; that is 0.5 hour, increase of its resource is difficult to considerably reduce it further. Therefore, it is suggested that not to give more resource to this activity.

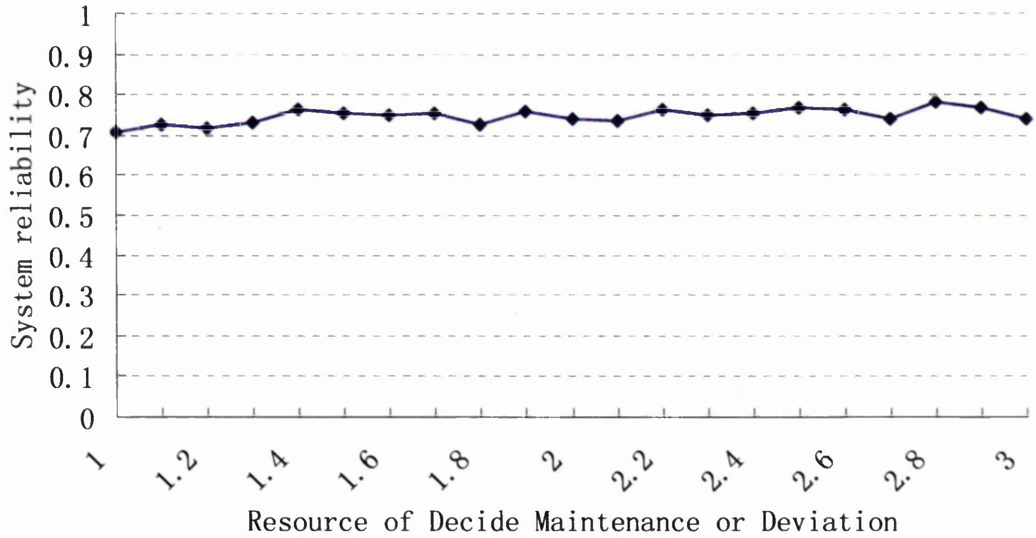


Figure 8.20: System Reliability against Resource of Decide Maintenance or Deviation

The activity 'Make Operational Proposal' did show a certain degree of influence on the system performance; however, the system reliability did not exceed 0.8; see figure 8.21. Thus, it can be concluded that a small quantity of resource can be added to the activity.

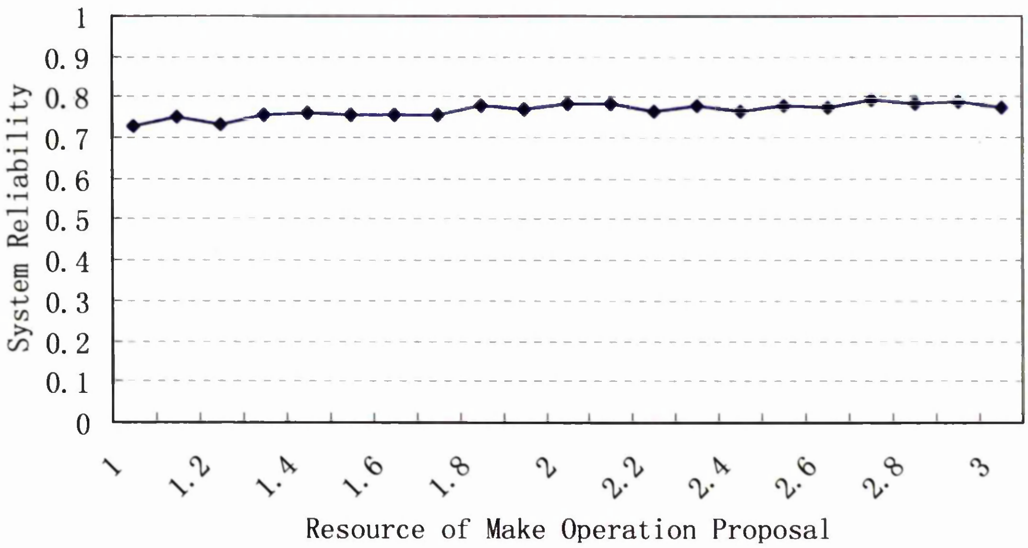


Figure 8.21: System Reliability against Resource of Make Operation Proposal

For the activity 'Decide Actions', the system reliability was obviously improved. However, it can be observed that there is no considerable improvement between the resources is 1.7 and 3; see figure 8.22; therefore, for the sake of economy, the resource given to the activity could be no more than 1.7 persons.

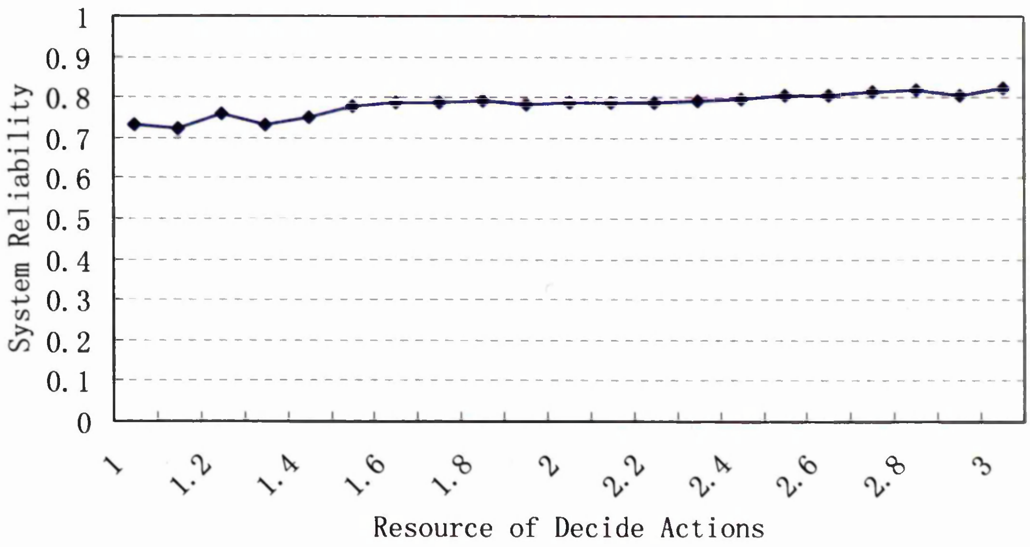


Figure 8.22: System Reliability against Resource of Decide Actions

A similar trend also happened to the activity 'Update Operational plan', 'Estimate Material Consumption' and 'Material Storage'; see figures 8.23, 8.24 and 8.25. From the resource is 1.8, the system performance hardly increased.

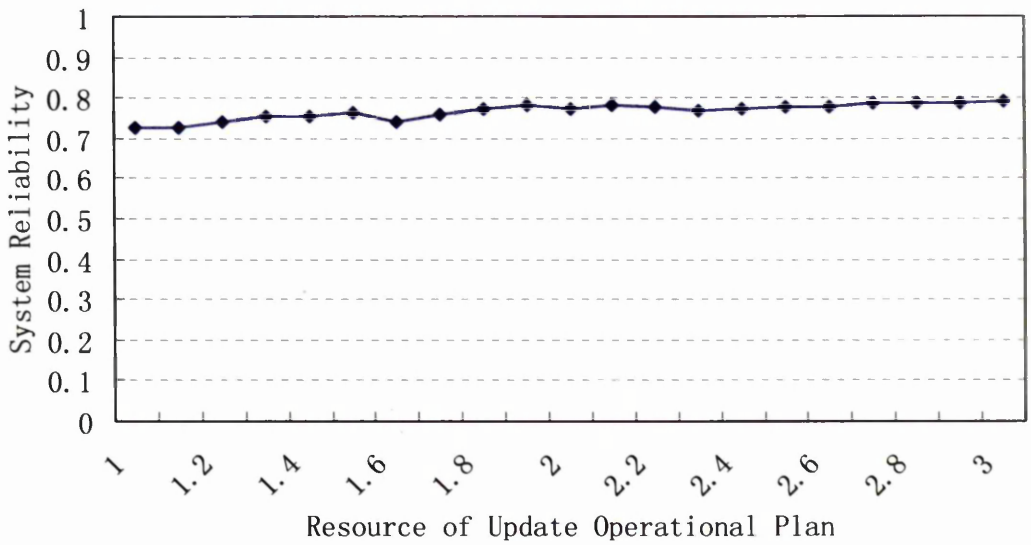


Figure 8.23: System Reliability against Resource of Update Operational Plan

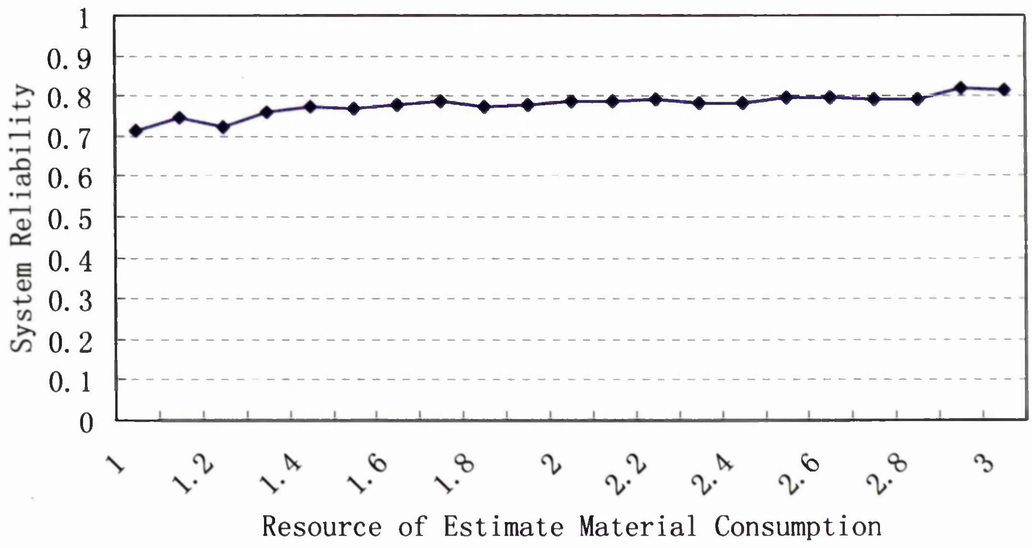


Figure 8.24: System Reliability against Resource of Estimate Material Consumption

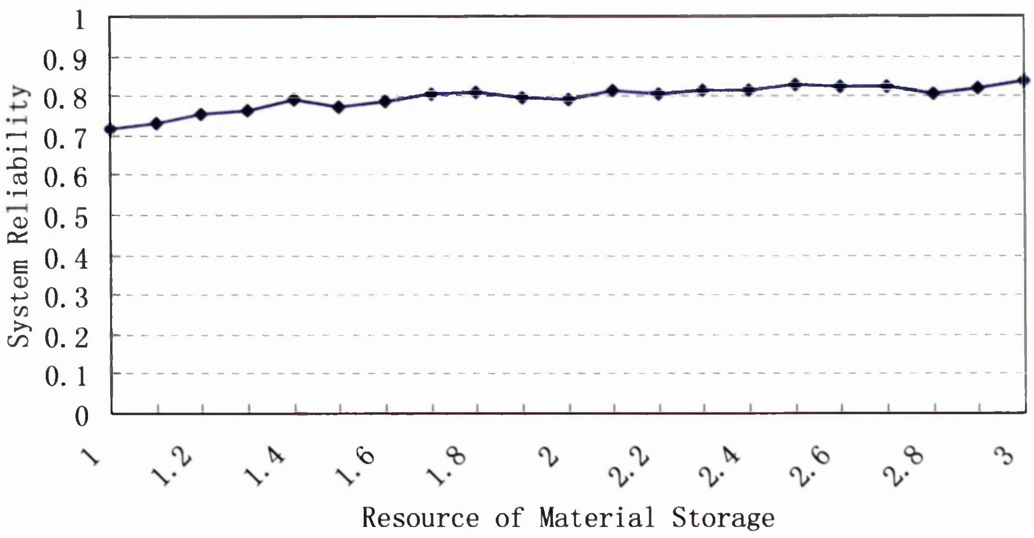


Figure 8.25: System Reliability against Resource of Material Storage

Among all activities, 'Transport to Flight Station' shows greatest impact on the system reliability. An increase of 2.3 times the basic resource causes the system reliability to increase to over 0.95; see figure 8.26. Evidently, system reliability is more sensible to the resource of 'Transport to Flight Station'. Thus, it is can be concluded that the most effective way to improve system reliability is reducing the time spent on transportation.

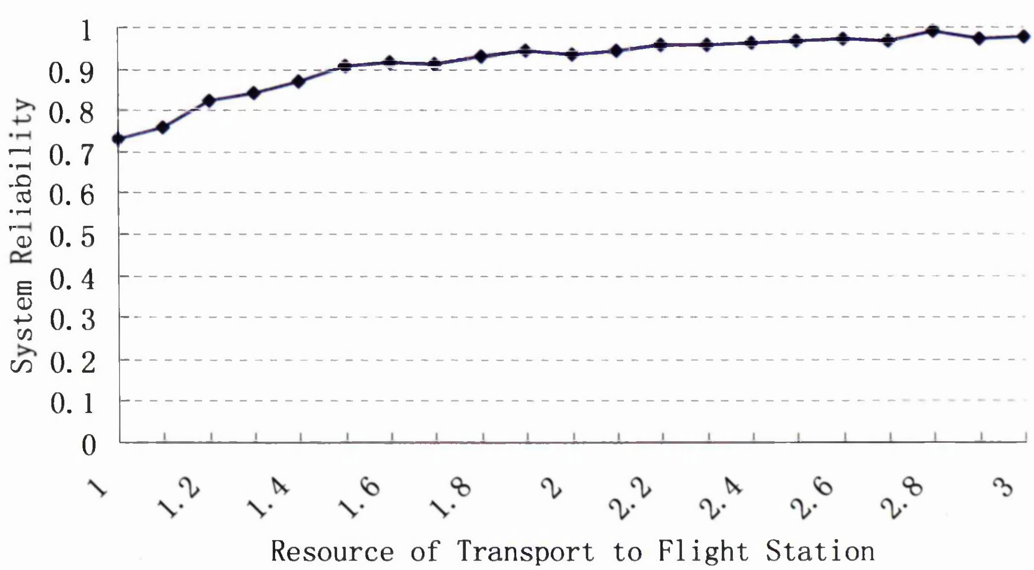


Figure 8.26: System Reliability against Resource of Transport to Flight Station

8.4.3 Sensitivity Analysis with respect to Reliability of Activities

The sensitivity of the system to the reliability of individual activities can also be explored. The reliability of each activity was varied from 0.5 to 1.0 and the effects on the system reliability were calculated. Figures 8.27, 8.28 and 8.29 show the results for activities respectively.

Generally, increases of the activity reliabilities would not bring the system performance to a very reliable status. The upper bound of that improvement seems to be 0.8. However, different levels of influence the activities have on the system performance can be shown.

The activities 'Verify Engine Failure' and 'Transport to Flight Station' show the greatest influence on the system reliability. The system reliability improves from below 0.5 to near 0.8 as the reliabilities of these two activities increase.

Comparatively, activities 'Judge Availability', 'Decide Maintenance or Deviation' and 'Estimate Consumption' seem have little influence on the system reliability. The reliability of the system does not improve obviously with the reliability increase of activities, which again reveals that this activity is not critical with respect to system reliability.

Other activities present moderate influence on system reliability.

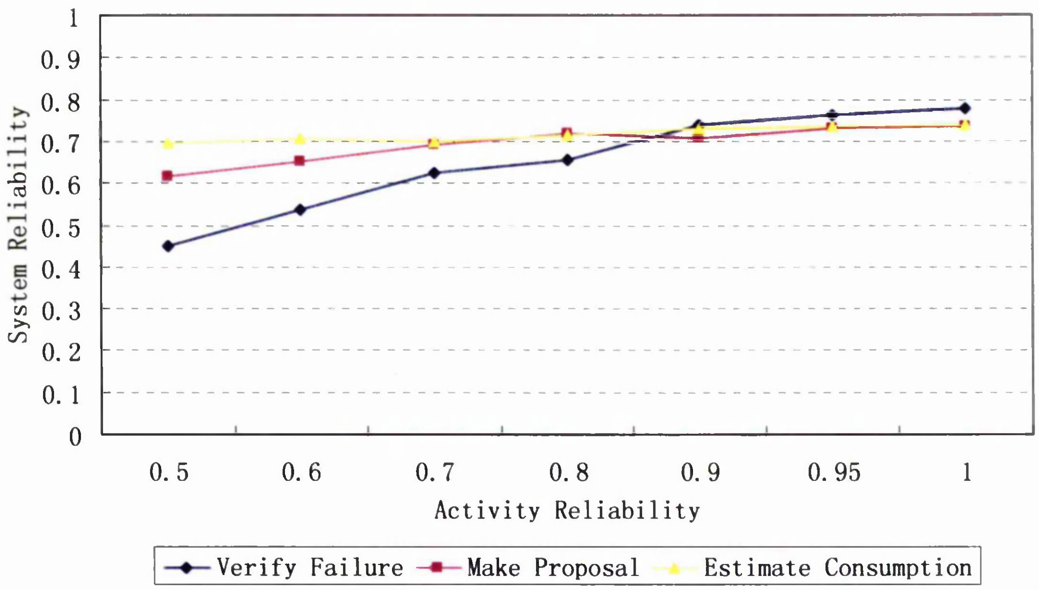


Figure 8.27: System Reliability against Reliability of Activities: Verify Failure, Make Proposal and Estimate Consumption

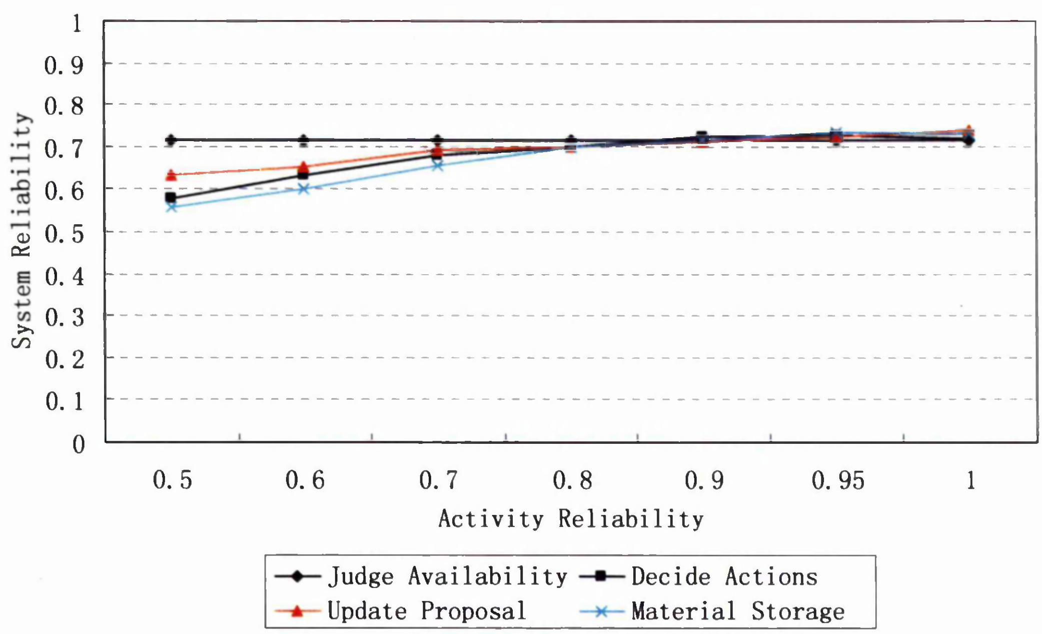


Figure 8.28: System Reliability against Reliability of Activities: Judge Availability, Decide Actions, Update Proposal and Material Storage

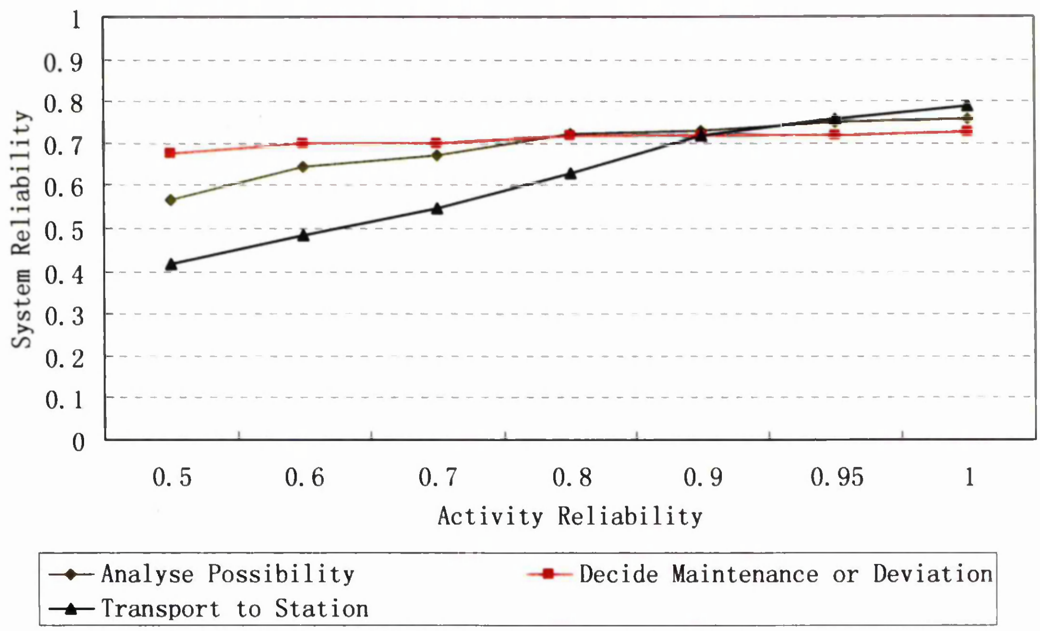


Figure 8.29: System Reliability against Reliability of Activities: Analyse Possibility, Decide Maintenance or Deviation and Transport to Station

8.4.4 Improved Use of Resource

The sensitivity analyses indicate that the resources of the activities 'Verify Engine Failure' and 'Transport to Flight Station' are critical to implement the contracted services successfully. Therefore, it is possible to improve system performance by giving more resources to them. Moreover, the resource given to the activity 'Judge Availability', 'Decide Maintenance or Deviation' and 'Estimate Material Consumption' could be reduced. A simulation has been run given the adjusted values displayed in the table 8.4 (the resource of the activities not included in the critical path remain unchanged).

The total resource used for all activities is kept constant with respect to initial system parameters given in the table 8.2. Therefore, system improvement is obtained by reallocation of resource and not by increasing the total resource used.

Activity \ Parameters	Resources (C_a)	Activity Time		Reliability (R_a)
		Mean (T_m)	Deviation (T_d)	
Receive notification of missing instructions	1	1/6	1/20	0.9
Verify Engine Failure	2	2	1	0.9
Register events	1	1/6	1/20	0.9
Judge Availability	0.25	2	0.1	0.9
Analyse Possibility for Maintenance or Deviation	1	1.5	0.5	0.9
Decide Maintenance or Deviation	0.5	1	1/6	0.9
Handle Deviation	1	0.5	1/6	0.9
Make Operation Proposal	0.5	2	1/3	0.9
Decide Actions	1	1.5	1/6	0.9
Update Operational Plan	1	1	0.2	0.9
Transport objects from flight station	1	7.5	1.5	0.9
Inspection and dismantle	1	2	1/3	0.9
Repair and assemble	1	5	1.5	0.9
Test and verify functionality	1	1	0.2	0.9
Store materials	1	2	1/3	0.9
Estimate Material Consumption	0.75	2	0.2	0.9
Seek and choose approved suppliers	1	1	0.2	0.9
Order materials	1	1.5	0.5	0.9
Control Received Materials	1	2	1/3	0.9
Transport to Flight Station	2	3.75	1.5	0.9

Table 8.4: Adjusted Values of Parameters of the SSS

The simulation results show that the system reliability can achieve 0.92 using the same resource (20 units); see figure 8.30. By reallocating the resource of the system, the system performance could be improved.

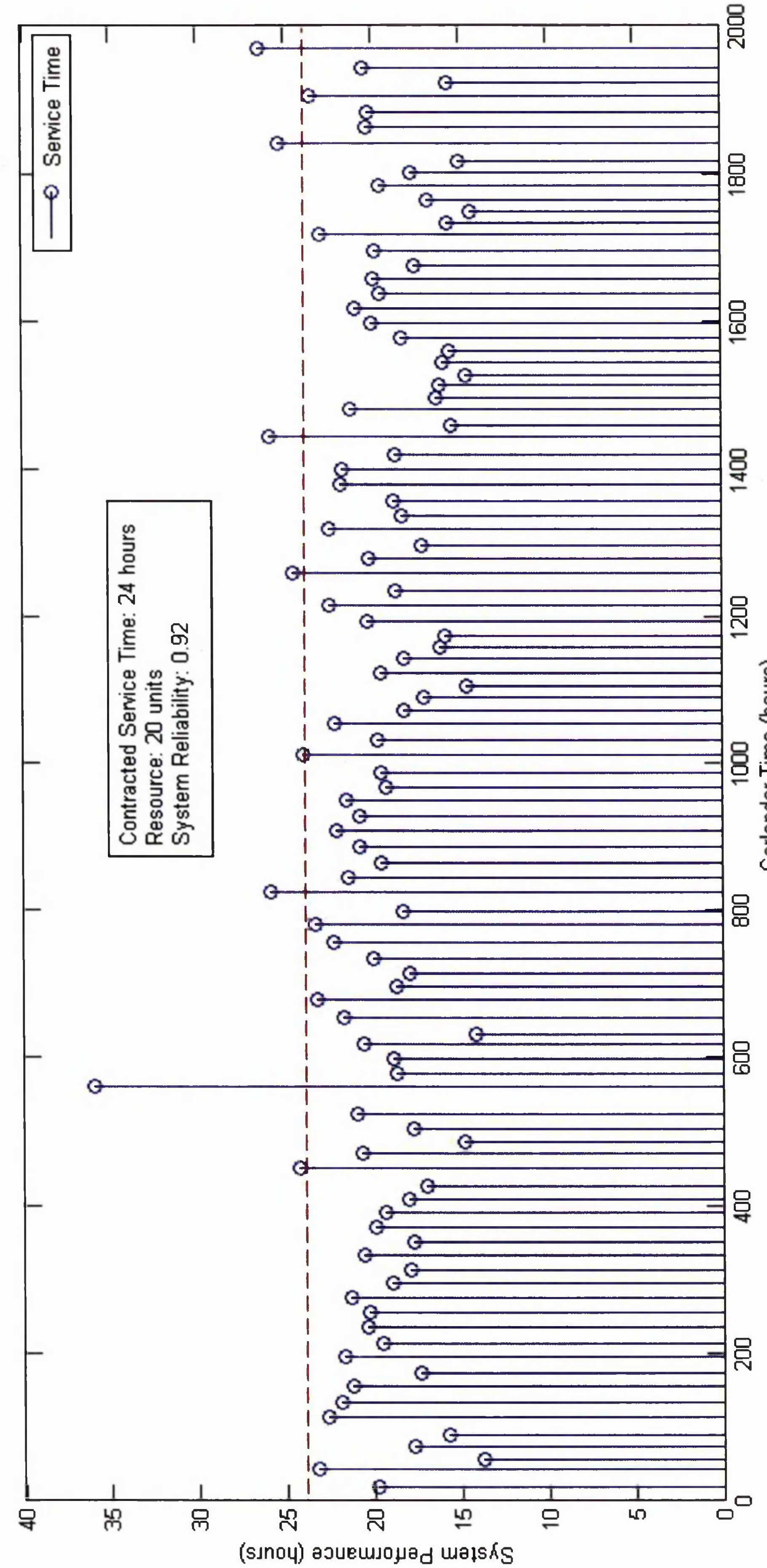


Figure 8.30: Improved System Performance with Same Resource

8.5 Summary

The 'top down' approach to system design has been used to create a system model for a service support system for the support of aircraft engines. A simulation of the system model has been carried out in order to identify those activities that are most critical to the effective operation of the system. By allocating resource to critical activities and reducing resource to others, the overall system performance may be improved without an increase in the total system resource. Also, the activities that are most sensitive to reliability have been identified; therefore, operations management can focus on those activities to ensure system performance.

Chapter Nine

Conclusions and Further Research

9.1 Conclusions

This research has been concerned with the design of service support systems and the reliability of the performance of the systems. A literature study on reliability was first carried out followed by a literature study on service support systems. It was found that the literature contained only general descriptions of services and that reliability assessment methodology had not been used quantitatively.

A study was made of industrial service systems to identify the particular activities of which they comprised. It was found that a system of defined activities, each with prescribed inputs and outputs, could be formed that would describe service support systems.

A literature review specifically concerning service design showed that similar generic steps were followed as in the case of hardware product design. However, such a generic approach could not lead easily to the creation of a functional system of linked activities. A system approach was needed. In this research, two system approaches to the design of services were derived: 'bottom up' and 'top down'. For customers who have a good understanding of this service system requirements, a 'bottom up' approach can be taken in which specific industrial activities can be defined and linked together to form a compatible total system. An alternative method is to start with a generic system that contains all essential activities plus some 'extras' and to customize that system to customer requirements by removing unwanted activities. Such an approach is suitable for customers who, perhaps because of their inexperience of service support systems, prefer to see a complete system and work from that starting point.

Service system failures were next considered which led to the consideration of internal and external system failures. Also, customer-perceived failures were recognized as being important in addition to contact failures. The way in which single activities 'fail' and their consequential influence on system performance is important. An activity may take longer than expected and it may have to be repeated. It is only the combination of the outcomes of the set of activities that compose that system that determines if the total service time takes longer than the contract specifies. Therefore, a system modelling and simulation approach was taken to the assessment of reliability.

A design approach based on the creation of a network of activities enables the system designers and customers to understand with clarity the precise composition of the service system and its functionality. This approach is also amenable to modelling and simulating; therefore reliability can be considered quantitatively at the design stage. A case study was carried out that showed how a service support system could be created using the system modelling approach proposed in this research. The case study also showed how, by simulating the system, activities that were critical to the reliable operation of the system could be identified and resources used efficiently.

9.2 Further Research

Functional products are integrated offers of hardware and a service support system. Therefore, evaluation of the reliability of the whole package should also consider the reliability of the hardware. Hardware reliability is a relatively well researched topic. Therefore, further research directions should consider the integration of the hardware and service system models, including hardware failure rates, number of pieces of hardware and a queuing system for particular service activities.

In this research, the reliability of all activities was modelled using a random number generator. No distinction was made between activities. However, the activities in a service system take a number of forms, some depend largely on human activity, some are computationally based and some include replacement of hardware components. It could be interesting in further research to take particular types of activities and to develop more sophisticated reliability models. For example, in the case of activities dominated by human activities, then human reliability analysis (Hollnagel, 2005) may be applicable. Such studies would not invalidate this research. Rather, their results could substitute the simple reliability model used in this thesis and so add value.

Two types of service failure have been identified: internal and external failures. The consequence of internal failures is increased service time and increased to the provider (not increased cost to clients). The consequences of external failures to the customers have not been developed in this research. There will be penalty clauses to the providers if the contract service time is exceeded, but clients may suffer deterioration in their reputation with their own customers. A further research topic could be to explore a more sophisticated experience of the consequences of external service failures. Therefore, by using the service modelling and simulation approach in this research to design a system and to give the probability of system failure, and deeper understanding of the consequences of failures, the business risk of service provisions may be analysed.

Reference

1. Alonso-Rasgado T., Thompson G. and Elfstrom B., 2004, *The Design of Functional (Total Care) products*, J.ENG.DESIGN, VOL. 15, NO.6, DECEMBER 2004, 515-540

2. Alonso-Rasgado T., Thompson G., Dannemark O.J., 2004b, *State of the Art in Service Design and Modelling*, VIVACE Public documentation URL: www.vivaceproject.com/content/engine/sdm_full.pdf

3. Alonso-Rasgado, T. and Thompson, G., 2006, *A rapid design process for Total Care Product creation*, Journal of Engineering Design, Vol. 17, No. 6, December 2006, 509–531

4. Akhtar, S., 1994, *Reliability of k-out-of-n: G Systems with Imperfect Fault-Coverage*, IEEE Transactions on Reliability, VOL. 43, No. 1, March 1994, 101-106.

5. Bums,S. and Gross,G., 1990, *VALUE OF SERVICE RELIABILITY*, IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 5, NO. 3, AUGUST 1990, 825-834

6. Bitran, G. and Pedrosa, L., 1998, *A Structured Product Development Perspective for Service Operations*, European Management Journal Vol.16, No. 2, 169-189

7. Bamford D. and Xystouri T., 2005, *A Case Study of Service Failure and*

Recovery within an International Airline, Managing Service Quality, Vol. 15 No. 3, 2005, 306-322

8. Bitner M.J., Booms B.H. and Tetreault M.S., 1990, *The Service Encounter: Diagnosing Favorable and Unfavorable Incidents*, Journal of Marketing, vol 54, no January, 1990, 71-84

9. Booz. Allen & Hamilton, 1967, *Management for New Products*, Booz, Allen & Hamilton Inc, Page 6

10. Burchill G. W. and Shen D., 1992, *Concept Engineering: The Key to Operationally Defining Your Customer's Requirements*. Center for Quality Management Cambrige, MA.

11. Cherng Cary, Andrew Lewis, and Simon Pai, 2005, *Modeling Toll Plaza Behavior Using Queing Theory*, MCM and ICM 2005, URL: http://www.math.washington.edu/~morrow/mcm/mcm.html

12. Chung B. and Hoffman K.D., 1998, *Critical Incidents: Service Failures That Matter Most*, Cornell Hotel and Restaurant Administration Quarterly, vol 39, no June, 1998, 66-71

13. Clark G., Johnston R. and Shulver M., 2000, *Exploiting the Service Concept for Service Design and Development*, In: Fitzsimmons, J., Fitzsimmons, M. (Eds.), New Service Design. Sage,Thousand Oaks, CA, pp. 71–91

14. Colgate M. and Norris M., 2001, *Developing a Comprehensive Picture of Service Failure*, International Journal of Service Industry Management, Vol.

12 No. 3, 2001, 215-233

15. Congram C. and Epelman M., 1995, *How to Describe Your Service*, International Journal of Service Industry Management, Vol. 6, no. 2, 1995, 6-23,

16. Cook, L. S., Bowen, D. E., Chase, R. B., Dasu, S., Stewart, D. M. and Tansil, D. A., 2002, *Human issues in service design*, Journal of Operations Management 20, 2002, 159-174

17. Dai Y.S, Xie M., Poh K.L., Liu G.Q., 2003, *A Study of Service Reliability and Availability for Distributed Systems*, Reliability Engineering and System Safety 79, 2003, 103–112

18. Davidson, J., 1994, *The Reliability of Mechanical Systems*, Mechanical Engineering Publications Ltd, UK ISBN 0-85298-881-8

19. DeVale, J., 1998, *Traditional Reliability*, Dependable Embedded Systems Spring, Carnegie Mellon University, 1998, 18-849b,
URL: http://www.ece.cmu.edu/~koopman/des_s99/traditional_reliability/

20. Dhillon, B.S., 2001, *Reliability Technology for Manufacturers: Engineering Better Devices*, Medical Device & Diagnostic Industry, (MDDI), Vol.23, No.10, 2001, 94-99

21. Edvardsson, B., 1992, *Service Breakdowns: A Study of Critical Incidents in an Airline*, International Journal of Service Industry Management, Vol. 3 No. 4, 1992, 17-29

22. Edvardsson, B. and Olsson, J., 1996, *Key Concepts for New Service*

Development, the Service Industries Journal; Apr 1996; 16, 2; 140-164

23. Fitzsimmons, J.A., 1998, *Service Management: Operation, Strategy and Information Technnology*, Boston, Mass. : Irwin/McGraw-Hill, c1998, ISBN: 0071157093

24. Flanagan, J.C., 1954, *The Critical-Incident Technique*, Psychological Bulletin, Vol. 51, No.4, July 1954, 327-358

25. Forbes, L.P., Kelley S.W. and Hoffman K.D., 2005, *Typologies of E-Commerce Retail Failures and Recovery Strategies*, Journal of Services Marketing, 19/5, 2005, 280-292

26. Goldstein, S. M., Johnston, R., Duffy, J. and Rao, J., 2002, *The Service Concept: the Missing Link in Service Design Research?*, Journal of Operations Management 20, 2002, 121-134

27. Griffin A. and Houser J. R., 1993, *The Voice of The Customer*, Marketing, Marketing Science, Vol. 12, 1, 1993, 1-27

28. Gronroos, C., 1988, *Service Quality: The Six Criteria Of Good Perceived Service*, Review of Business; Winter 1988; 9, 3; 10-13

29. Guinet, A., 1990, *Knowledge acquisition and assessment about production management systems*, European Journal of Operations Research, Vol. 45, April 1990, 265-74

30. Harrison, D. K. and Petty, D. J., 2002, *Systems for Planning and Control in Manufacturing*, Oxford: Newnes

31. Hill, A. V., Collier, D.A., Froehle, C.M., Goodale, J.C., Metters, R.D. and Verma, R., 2002, *Research Opportunities in Service process Design*, Journal of Operations Management 20, 2002, 189-202

32. Hollins, B., 2006, *Service Design*, Design Council, URL: http://www.designcouncil.org.uk/en/About-Design/Design-Disciplines/Service-design-by-Bill-Hollins/

33. Hollnagel, E., 2005, Human Reliability Analysis, URL: http://www.ida.liu.se/~eriho/HRA_M.htm,

34. Huang, D. and Billinton, R., 2007, *Repair State Residence Time Distribution Considerations In Electric Power Generating Capacity Adequacy Evaluation*, Advances in Risk and Reliability Technology Symposium, Loughborough University 2007, 307-318

35. International Organisation for Standardisation (ISO, 9004-2), 1991, Quality Management and Quality Systems Element- Part 2 Guidelines for services

36. Johne, A. and Storey, C.,1997, *New Service Development: A review of the Literature and Annotated Bibliography*, European Journal of Marketing 32, 3/4, 1997, 184-251

37. Karwan, K. R. and Markland, R. E., 2006, *Integrating service design principles and information technology to improve delivery and productivity in public sector operations: The case of the South Carolina DMV*, Journal of Operations Management 24, 2006, 347–362

38. Kelley S.W., Hoffman K.D. and Davis M.A., 1993, A Typology of Retail Failures and Recoveries, *Journal of Retailing*, vol 69, no Winter, 1993, pp

429-452

39. Kotler, P., 1983, *Principles of Marketing*, the Prestice-Hall series in marketing, Englewood Cliffs, London: Prentice-Hall, c1983, ISBN: 0137016239

40. Lovelock, C.H., 1984, *Developing and Implementing New Services*, in George, W.R. and Marshall, C.E. (Eds), Developing New Services, American Marketing Association, Chicago, IL, 44-64.

41. Ittig, P.T., 1994, *Planning Service Capacity When Demand Is Sensitive to Delay*, Decision Sciences, Jul/Aug 1994, 25, 4; 541-557

42. Marca, D.A., 1991, *Augmenting SADT to Develop Computer Support for Cooperative Work*, International Conference on Software Enginering, Austin, Texas, 1991, 94-103

43. Menor, L. J., Tatikonda, M. V. and Sampson, S. E., 2002, *New service development: areas for exploitation and exploration*, Journal of Operations Management 20, 2002, 135–157

44. Meuter M.L., Ostrom A.L., Roundtree R.I., & Bitner M.J., 2000, *Self-Service Technology: Understanding Customer Satisfaction with Technology-Based service Encounters*, Journal of Marketing, Vol.64, July 2000, 50-64

45. Murphy, P. E. and Robison, R. K., 1981, *Concept Testing Services, In Marketing of Services*, Proceedings Series of the American Marketing Association, 217-220

46. Nytomt, F., 2004, *Service Reliability and Maintainability*, Licenciate Thesis, Lulea University of Technology, Lulea, Sweden

47. O'Connor, P. D. T., 1991, *Practical Reliability Engineering*, John Wiley & Sons Ltd., UK ISBN 0-471-92902-6

48. Persson, F., 2004, *Service Concept Design*, Licentiate Thesis, Lulea University of Technology, Department of Applied Physics and Mechanical Engineering, Division of Computer Aided Design, 2004:56, ISSN: I402-I757

49. Roberto Gilioli Rotondaro and Claudio Lopez de Oliveira, 2001, *Using Failure Mode Effect Analysis (FMEA) to Improve Service Quality Service Operations Management*, Proceedings of the Twelfth Annual Conference of the Production and Operations management Society, POM-2001, march 30-April 2,2001, Orlando Fl.

50. Scheuing, E.E. and Johnson, E.M., 1989a, *New product development and management in financial institutions*, International Journal of Bank Marketing, Vol. 7 No. 2,17-21

51. Shostack G.L., 1981, *How to Design a Service*, European Journal of Marketing, Vol.16 No.1, 49-63

52. Shostack G.L., 1984a, *Designing Services that Deliver*, Harvard Business Review, January-February 1984,133-139

53. Shostack, G. L., 1984b, *Service Design in the Operating Environment*, In Developing New Services, edited by W.R. George and C.E. Marshall (Chicago, IL: American Marketing Association), 27-43

54. Siltala M., 2004, *Modelling Contracting Procedure and the Concept of the Service Portfolio for Finnish Municipalities using SADT Technique*, Nordic Journal of Surveying and Real Estate Research, VOL 1, 2004

55. Smith, D. J., 1997, *Reliability, Maintainability and Risk*, Butterworth-Heinemann Ltd, UK ISBN 0-7506-3752-8

56. Spreng, Richard A; Harrell, Gilbert D; Mackoy, Robert D, 1995, *Service Recovery: Impact on Satisfaction and Intentions*, The Journal of Services marketing; 1995; 9, 1; pg.15

57. Stefan M., 2003, *Perceived Justice in Service Failure and Recovery Incidents: A Multiple Correspondence*, Approach AMA Winter Educators' Conference 2003

58. Sze, D., 1984, *A Queuing Model for Telephone Operator Staffing*, Operations Research, Mar/Apr 1984, Vol. 32 Issue 2, 229-249

59. Taylor, S., 1994, *Waiting for Services: The Relationship between Delays and Evaluations of Service*, Journal of Marketing, Vol.58, April 1994, 5 6-69

60. Thompson, G., 1999, *Improving Maintainability and Reliability through Design*, Professional Engineering Publishing Ltd., UK ISBN 1-86058-135-8

61. Tortorella, M., 2005, *Service Reliability Theory and Engineering*, Quality Theory and Quantitative Management, vol. 2, no. 1, 2005,1-37

62. Travel & Tourism, 2003, Vol. II No. 2 Saturday 11 January - 17 January

2003, URL: http://www.chiangmai-mail.com/012/tours.shtml

63. Ulrich K. T. and Eppinger S. D., 1995, *Product Design and Development*, McGraw Hill Co, New York, NY, ISBN: 0070658110

64. Verma, R., Fitzsimmons, J., Heineke, J. and Davis, M., 2002, *Editorial: New issues and opportunities in service design research*, Journal of Operations Management 20 (2002), 117–120

65. Von Hippel E., 1986, *Lead User: A Source of Novel Product Concepts*, Managements Science, Vol 32-7, 1986, 791-805.

66. Walker, B. K., Wereley N. M., Luppold R. H., and Gai E., 1989, *Effects of Redundancy Management on Reliability Modeling*, IEEE Transactions on Reliability, VOL. 38, NO. 4, 1989 October 475-482.

67. Weissberger A.J., 2004, *Web Services Reliability Becomes Oasis Standard*, DAILY NEWS AND INFORMATION FOR THE GLOBAL GRID COMMUNITY / NOVEMBER 15, 2004: VOL. 3 NO. 46

68. Wheelwright S. and Clark K. B., 1992, *Revolutionizing Product Development: Quantum Leaps in Speed, Efficiency and Quality*, The Free Press, New York.

69. Wittwer, J.W., 2004, *Monte Carlo Simulation Basics*, from Vertex42.com, June 1, http://vertex42.com/ExcelArticles/mc/MonteCarloSimulation.html

Appendix A

System Design Interface

This appendix illustrates the use of the system design interface (SDI) during the 'bottom up' design process. In addition, the programming codes for the system compatibility check are presented.

A.1 Layout of the SDI

The SDI is an application programmed with Visual Basic and aims to facilitate the 'bottom up' service design approach especially the system compatibility check process. It displays the system under development in a hierarchical way and the root level is as shown in the figure A.1.

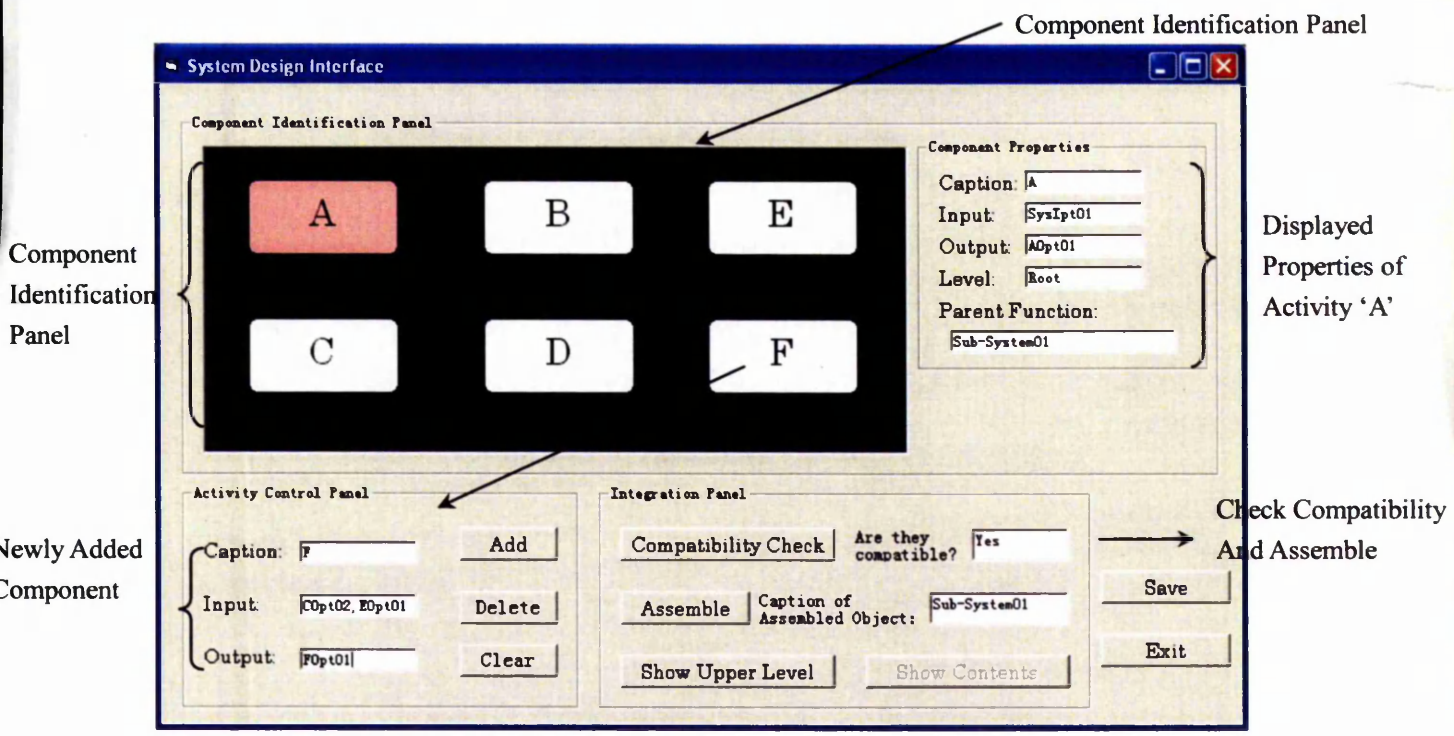


Figure A.1: Root Level of the Graphic User Interface developed in Visual Basic

The root level comprises three main parts and they are:

1. Component Identification Panel
2. Activity Control Panel
3. Integration Panel

A.2 Component Identification Panel

The component identification panel is mainly used to visualize the identified activities and to display their properties; see figure A.2.

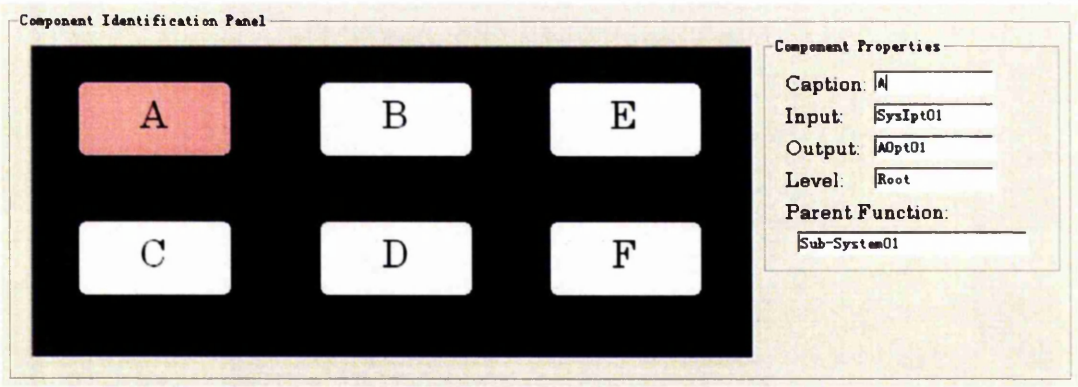


Figure A.2: Component Identification Panel

The properties of an activity include:

1) Caption ---- name of the activity;
2) Input ---- required inputs to carry out the activity;
3) Output ---- outputs generated by the activity;
4) Level ---- the level the activity stays; e.g. 2^{nd}, 3^{rd} etc.;
5) Parent Function ---- the sub-system or the system the activity belongs to.

These properties could be edited by simply changing the displayed information in the text boxes.

A.3 Activity Control Panel

This panel is mainly used to modify the system under development. Users can add more activities to the system or delete unwanted ones; see figure A.3.

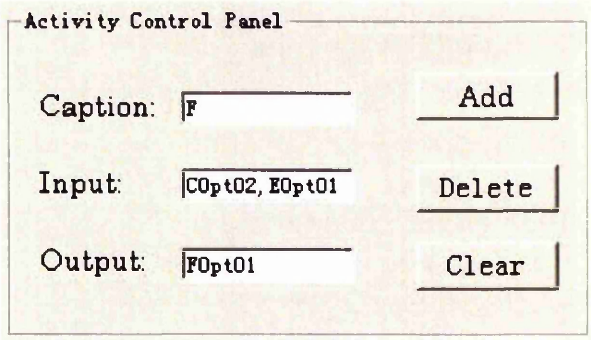


Figure A.3: Activity Control Panel

The functions of the buttons are:

1) Add ---- add further identified activities;

2) Delete ---- delete unwanted activities;

3) Clear ---- clear all activities in the component identification panel.

A.4 Integration Panel

The main purpose of this panel is to check the compatibility of the system under development; see figure A.4.

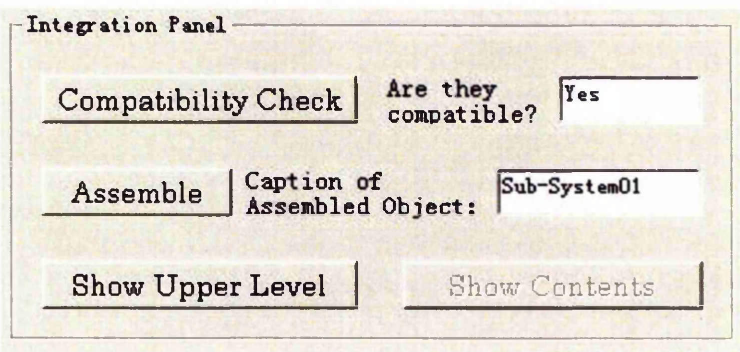


Figure A.4: Integration Panel

With this panel, users can:

1) check the compatibility of the system;
2) Assemble compatible components as a element of higher level;

A.5 Programming codes for System Compatibility Check

System compatibility check is a main purpose of the SDI. The programming codes for the function are as below:

```
Function compareArray(ByRef A() As String, ByRef B() As String) As Integer
Dim i As Integer, j As Integer, iCount As Integer

compareArray = 0 'default as not equal
If UBound(A) = UBound(B) Then

For i = 1 To UBound(A)
    iCount = 0
    For j = 1 To UBound(B)
        If A(i) = B(j) Then
            iCount = iCount + 1
        End If
    Next j
        If iCount = 0 Then
            Exit Function
        End If
Next i
If iCount = 0 Then
            Exit Function
        End If
compareArray = 1 'reset the value as equal
Else
compareArray = 0
End If
End Function
```

Table A.1: Programming Code of System Compatibility Check: Part 1

```
Private Sub Command_CPcheck_Click()
Dim Output_Available(100) As String, Input_Required(100) As String
Output_Available(1) = "a"
Output_Available(2) = "c"
Output_Available(3) = "c"
Input_Required(1) = "a"
Input_Required(2) = "b"
Input_Required(3) = "c"

If compareArray(Input_Required, Output_Available) = 1 Then
Text_CP.Text = "System Compatible"
Else: Text_CP.Text = "System Inconsistent"
End If
End Sub
```

Table A.2: Programming Code of System Compatibility Check: Part 2

Appendix B

Simulation Models of Typical Activities

This appendix presents how the four typical activities (see section 3.3) are modelled using the approach introduced in the chapter 7. Creating models for these four activities forms the basis of building the simulation model of SSSs.

Type	Descriptions
Type 1	Activity with Single Input and Output
Type 2	Activity with Multiple Inputs
Type 3	Activity with Multiple Outputs
Type 4	Activity with Multiple Inputs and Outputs

Table B.1: Typical Activities in SSSs

B.1 Activity with Single Input and Output

This section depicts consecutive steps an activity with single input and output carries out. As shown in the figures B.1, activity A has two states *Standby* and *Working* which are tagged with 0 and 1. In the step one, activity A enters into the standby state until the required input is available.

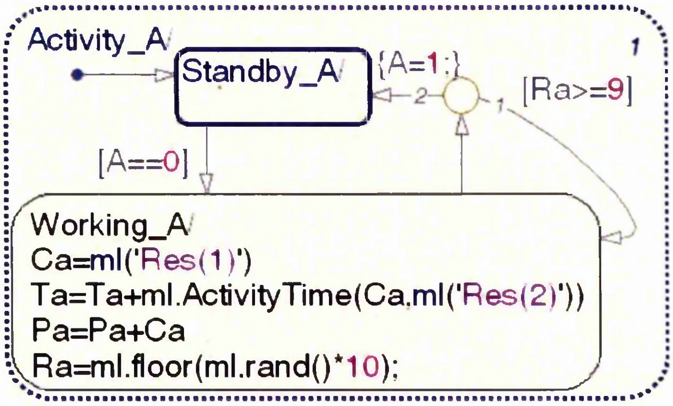


Figure B.1: Step one of carrying out Activity A

If the input is ready, e.g. **[A==0]** is true, the activity goes to the working state; see figure B.2.

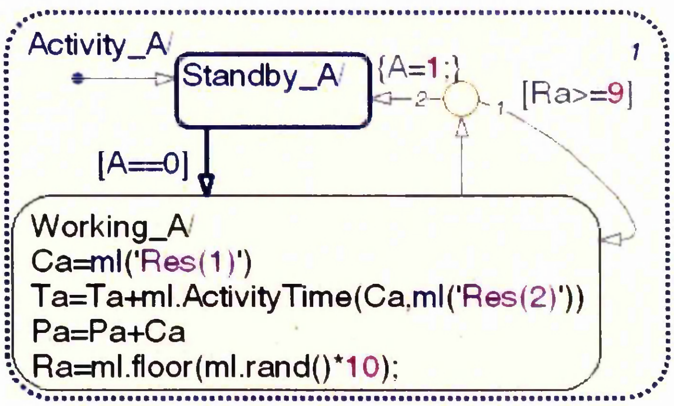


Figure B.2: Step two of carrying out Activity A

After a period of time t_a, Activity A finishes and its output is about to be checked. If, the output is not satisfactory, the activity will be repeated until it's qualified and then the activity generates output **{A=1;}** and goes back to the standby state; see figures B.3 and B.4.

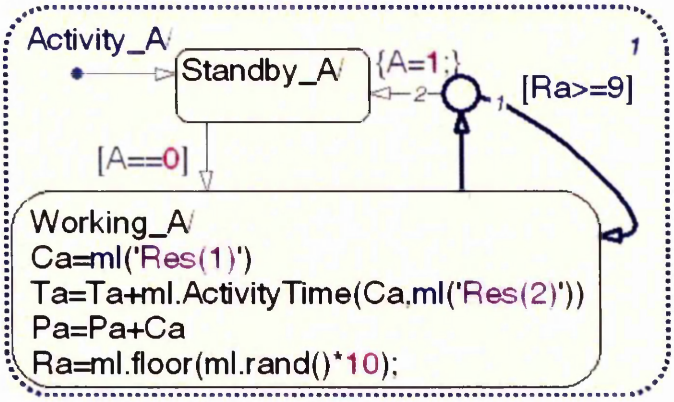


Figure B.3: Step three of carrying out Activity A

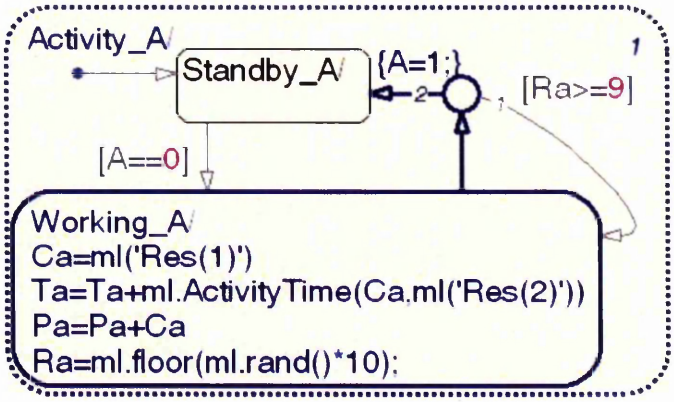


Figure B.4: Step four of carrying out Activity A

Values of the time and resources consumed by the activity are stored by T_a and P_a respectively.

B.2 Computational Model of Activity with Multiple Inputs

For an activity of this type to carry out all it required inputs must be ready; therefore, before it enters into the working state, an input checking mechanism is needed. A variable indicating if all required inputs are ready needs to be defined and its value is returned by the MATLAB embedded function '*Truthtable*'.

For the two sub-classes in this group: Activity with Multiple Mandatory Inputs and the Activity with Multiple Selected Inputs, they can be modeled in the same appearance; the only difference is the content of the '*Truthtable*'. For example, see figure B.5, the activity A is expecting three inputs which are denoted by X, Y and Z. A variable 'A_Ready' is defined to indicate the state of inputs. If the activity belongs the former, the value of 'A_Ready' returned by the '*Truthtable*' will be 'True' if X, Y and Z are all ready; otherwise, it returns the value of 'A_Ready' as 'true' given any one or one combination of them is ready.

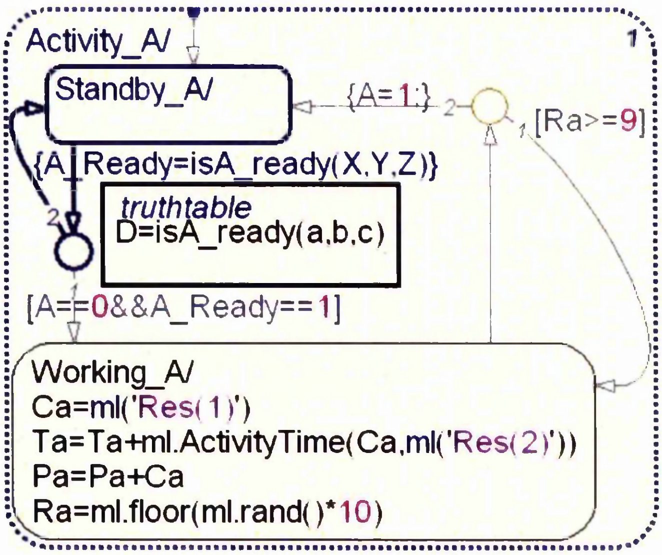


Figure B.5: Waiting for Inputs: first step of carrying out an activity with multiple inputs

Figure B.5 shows that the inputs that required by the activity A is not met; therefore, the activity returns to the standby state.

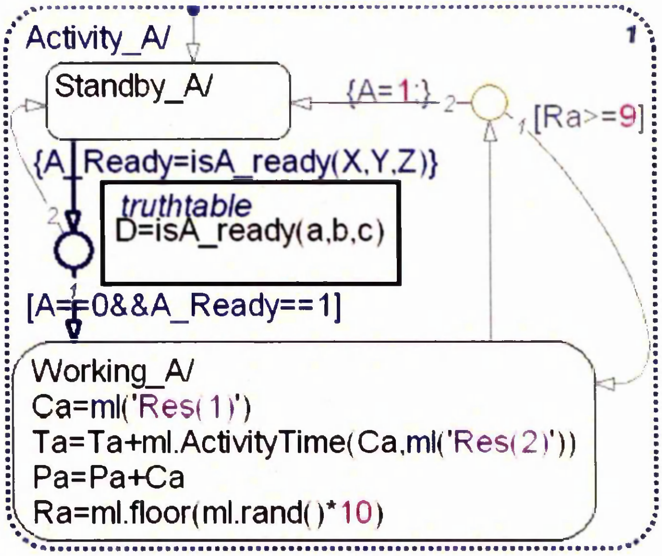


Figure B.6: Passing the Judge Point

And when all the conditions have been met, the state of the activity will transfer immediately, see figure B.6. The rest steps of modelling the activity are in line with those introduced in the last section; therefore it will not be unnecessarily repeated.

B.3 Computational Model of Activity with Multiple Outputs

Activities of this type probably generate more than one output. Outputs could be mutually exclusive or in parallel depending on the inherence of the activity. Before sending outputs, the output(s) will be checked, none of them will be delivered should the activity have been implemented poorly, see figure B.7.

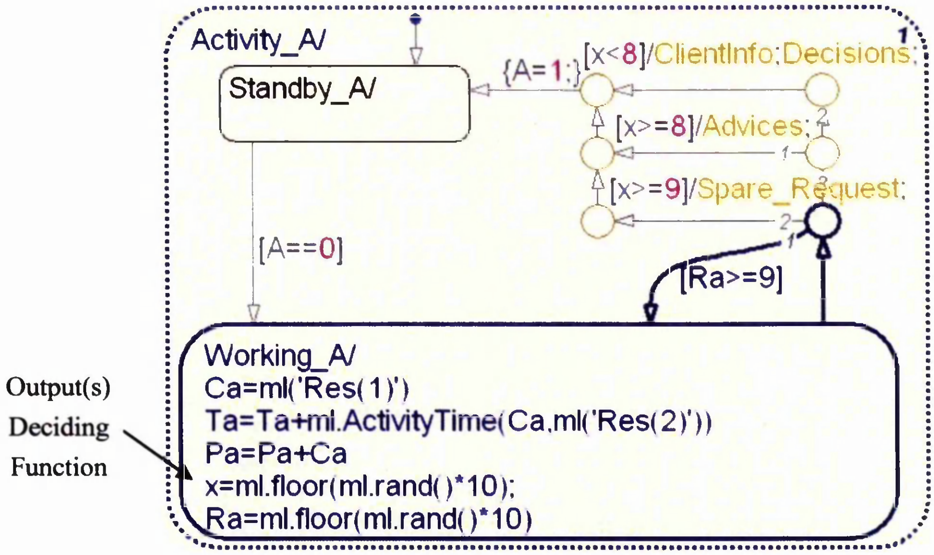


Figure B.7: Repeat of Activity with Multiple Outputs

Figure B.7 illustrates that the activity A outputs different information based on the value of variable '***X***'. The activity contains a hybrid case of the two sub-groups and its outputs are displayed in the table B.1.

The model uses a MATLAB embedded function '**x=ml.floor(ml.rand()*10)**' as an example of the output deciding function and it creates a random number between 0 and 10 as the value of '***X***'. In other cases, other appropriate functions may be used.

Value of '*X*'	Outputs
'*X*'>=9	Spare Request;
'*X*'>=8	Advices;
Others	Client Information; Decisions;

Table B.2: Outputs of Activity A

If '***X***' >=9, the model output a '**Spare Request**' which is mutually exclusive with '**Advices**' and a combination of '**Client Information**' **and** '**Decisions**'; see figure B.8.

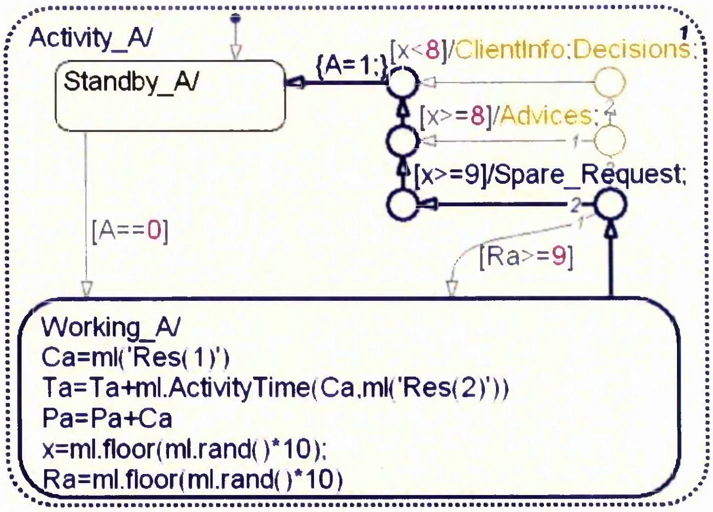


Figure B.8: Selected Output of Activity with Multiple Outputs

In some case, parallel outputs may be generated. For example, should ***'X'*** is smaller than 8, the activity output '**Client Information**' **and** '**Decisions**' simultaneously; see figure B.9.

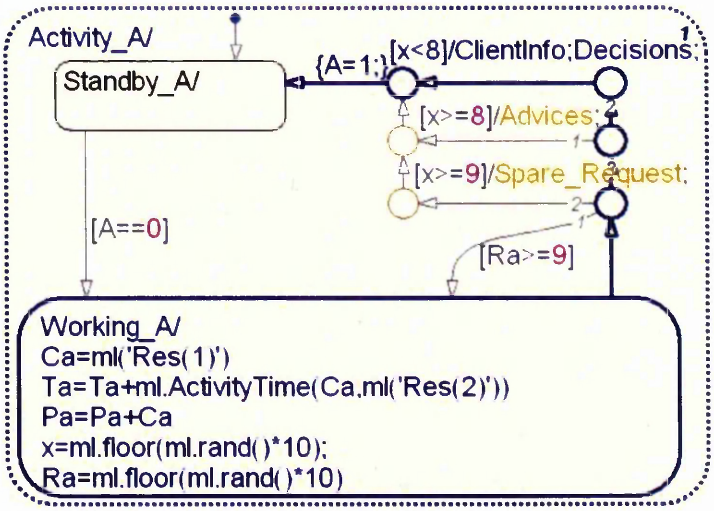


Figure B.9: Parallel Outputs of Activity with Multiple Outputs

B.4 Computational Model of Activity with Multiple Inputs and Outputs

Activities of this sort are hybrids of type 2 and type 3. Simulation models contain a '*Truthtable*' and an output deciding function; see figures B.10 and B.11.

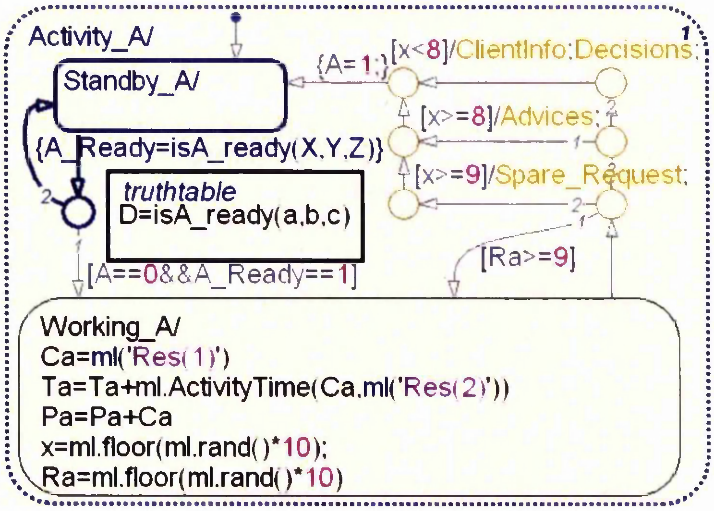


Figure B.10: Input Checking of Activity with Multiple Inputs and Outputs

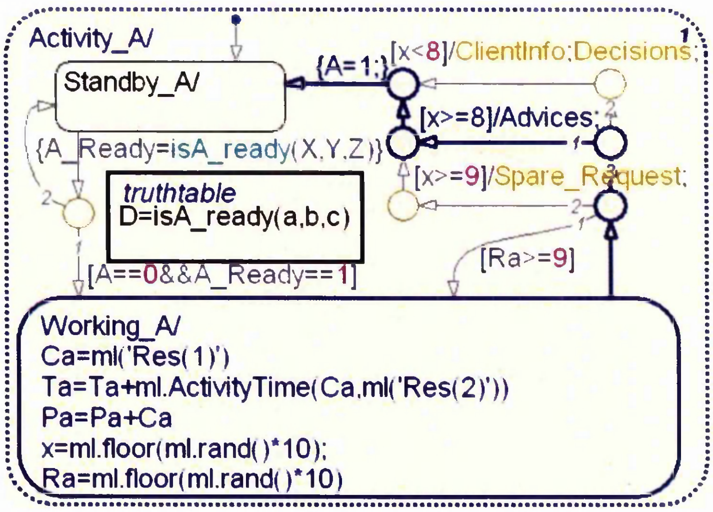


Figure B.11: Generating Output

Appendix C

Publication

The Failure of Service Support Systems

Proceedings of the 17th Advances in Risk and Reliability Technology Symposium

Author: G. Thompson, G. Wang and T. Alonso-Rasgado

School of Mechanical Aerospace and Civil Engineering

University of Manchester, UK

Abstract

The paper is concerned with the failure of service support systems which are part of a Functional (Total Care) Product. A brief description of such a system is first given followed by a literature review of service failures. Systems may fail with respect to contractual performance indicators and, importantly, with respect to customer expectations. Internal and External system failures are identified. Internal failures incur increased costs but do not affect external performance to a degree which violates contracted performance requirements. External failures are all those failures, contracted and perceived, of which the customer is aware. The discussion leads to a study of the reliability of the system as a function of individual activity performances.

1. Introduction

This paper is concerned with the provision of Functional Products, that is, products that include hardware and support services. Another name used in industry for such products is Total Care products. Examples include the sale of 'power by the hour' to aircraft operators rather than the sale of the aircraft engine hardware, and the supply and support of key process machinery that perform a defined function as part of a production process rather than the sale of the equipment. Functional products are significantly different in character to the traditional 'hard' product and the seller of the function must consider the reliability of all parts of the functional product, both the hardware and the service system. Hardware reliability has received much attention in the literature, but service reliability less so.

One advantage to the user of functional products is a guarantee of availability. The advantages to the supplier are the intimate long term, stable business to business relationships and the product knowledge gained through operation. The reliability of the service system is a key aspect of the successful provision of a functional product.

The paper first gives an outline description of a service support system using an example in the open literature, followed by a literature review of service failures. Customer-perceived failure in contrast to contractual failure, and internal and external system failures are next discussed. The paper concludes with a modelling and simulation approach to the analysis of service systems to determine 'system failure'.

2. Service Support Systems

Services vary in form from industry to industry but they share some common characteristics that make them special in contrast to conventional hardware products. Shostack (1984) made a widely accepted statement: 'services are unusual in that they have impact, but no form .... they can't be physically stored or processed and their consumption is often simultaneous with their production.' Gronroos C., (1988) further summarized four basic four basic characteristics of services:

'Services are intangible

Services are activities or a series of activities rather than things.

Services are at least to some extent produced and consumed simultaneously.

The customer participates in the production process at least to some extent.'

Figure 1 shows some key elements of a service support system. Such systems are multi-disciplinary. Hill (2002) describes a wide range of opportunities for improved service design, including service design for manufacturing. The introduction of new technologies and the development of integrated supply chains are seen as factors that increase the importance of providing effective services. The services add value to the customer's use of the tangible product and lower the total life cycle cost of the product.

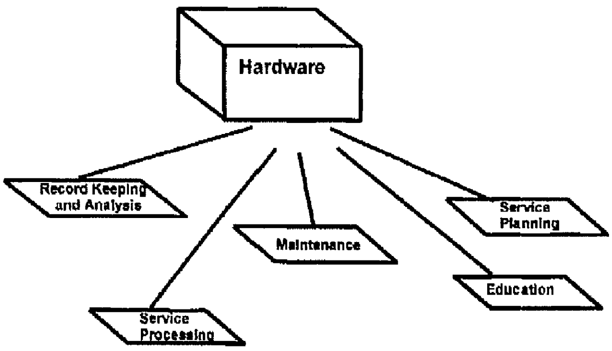


Figure 1: Elements of a functional product (Alonso-Rasgado et al., 2004)

A service support system is extremely complicated. Figure 2 shows a service support system for a Total Care System for aircraft engines. The inputs of the system include data, hardware and resources and the output is a functional engine. Each activity may be defined by a process diagram, Figure 3 shows a typical example.

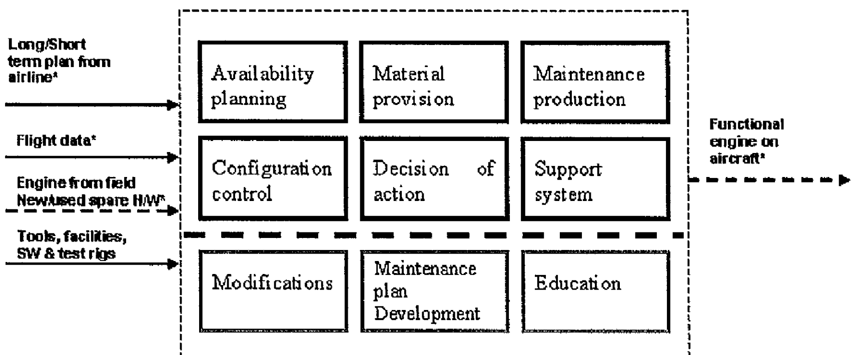


Figure 2: Functional level of Service Support System (Adapted from Persson, F., 2004).

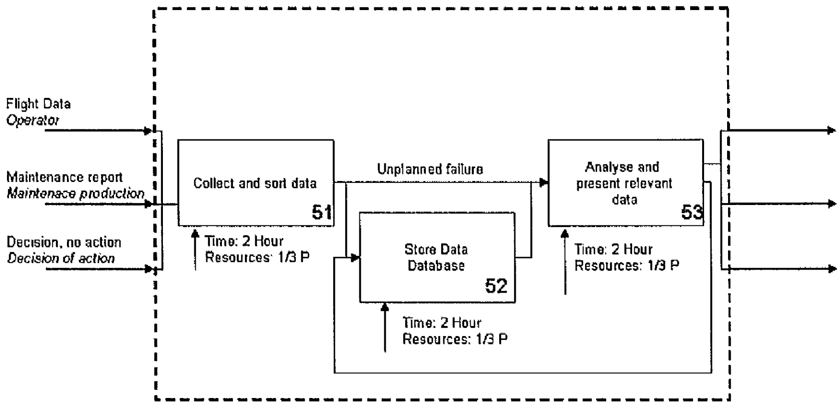


Figure 3: Activity level (Persson, F., 2004)

3. Service Failure

3.1 General Service Sector Failures

Gronroos (1988) identified reliability as one of six criteria in the assessment of service quality. However little research has since been done on service reliability but much effort has been given to service recovery strategies. This is because, in the general service industries, recovery actions tend to be effective and inexpensive to implement. For instance, the average recovery rate is about 8 on a scale of 1 (very poor) and 10 (very good) (Kelley *et al.*, 1993). The rapid growth of complex modern engineering systems in which there are more integrated technology elements leads to a high cost of recovery. Also, in many engineering applications safety problems may occur if the support system fails to keep equipment operating in good order. Therefore, for engineering service support systems, the emphasis should be on creating reliable systems rather than recovery strategies.

Service incidents have been collected by researchers using the Critical Incident Technique (CIT) (Flanagan J.C., 1954). During the last decade incidents have been reported from service sectors such as restaurant, retailing and airlines (Bitner *et al.*, 1990; Edvardsson, 1992; Kelley *et al.*, 1993; Chung *et al.*, 1998, Meuter *et al.*, 2000, Forbes *et al.*, 2005;). Terms such as 'service errors' and 'unfavourable events' are used rather than 'service failure'.

There exist two types of service failures: contract failure and customer perceived failure. A review of the general failure modes of a wide range of service sectors reveals that the majority of service failures are contract failures. As reported by Edvardsson, (1992), *Delayed* and *Cancelled Flights* account for 82 per cent of all the negative incidents. In restaurants, contract failures account for more than 70 per cent of all failures. (Chung *et al.*, 1998).

Customer perception is a key factor though and the evaluation of the severity of a service failure is more or less subjective (Kelley *et al.*, 1993, Chung and Hoffman, 1998). Service failure sometimes can have a positive influence on the relationship between providers and customers if the recovery process is perceived as good, that is, the customer feels valued by the service provider. Service failures can be recovered better than expected but failures with bad recoveries tend to be memorable (Stefan, 2003). Human behavior is, as would be expected, an important source of service system failures (Bitner *et al.*, 1990)

3.2 Failure Characteristics of Engineering Service Support Systems

3.2.1 Contract Failures and Customer Perceived Failures

A contract for service support system for a Functional (Total Care) Product has Key Performance Indicators (KPIs) that define the level of service to be provided. The level of performance of each KPI may be monitored with time. Any fall below the contracted level of performance may be defined as a contract failure, as shown in figure 4.

However, should the actual performance be well above the contracted performance for long periods, then customer expectations are raised. Any fall in actual performance below the expectation of performance is then perceived as a failure, even though the actual performance is not below contracted performance, see figure 5. Though customer perceived failures will not incur financial penalties on the total care product providers, they do harm the long term relationship with customers.

The optimum system performance with respect to the contract is one with low standard deviation and minimum 'safety margin' above the contracted performance, see figure 6. Thus customer expectation and satisfaction are managed well at minimum cost, i.e. resources are not used to generate performance well above that contracted.

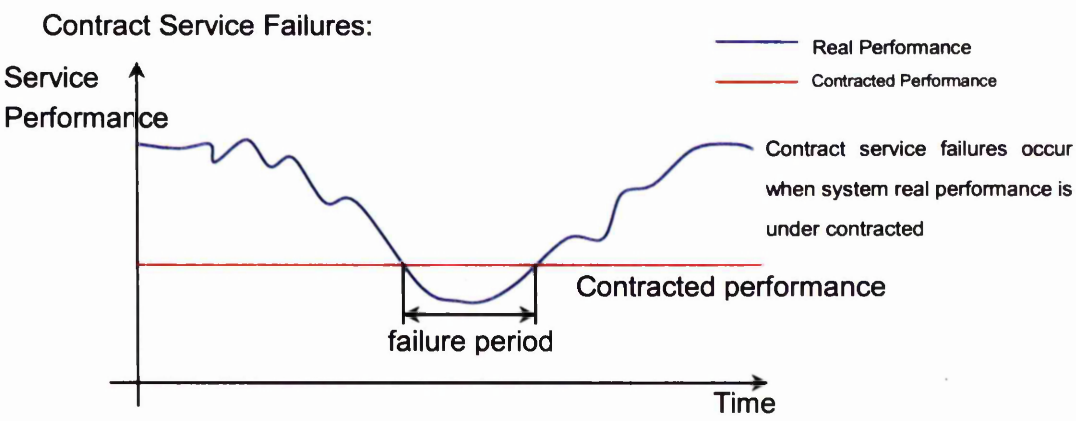


Figure 4: Contract Service Failures

Perceived failures:

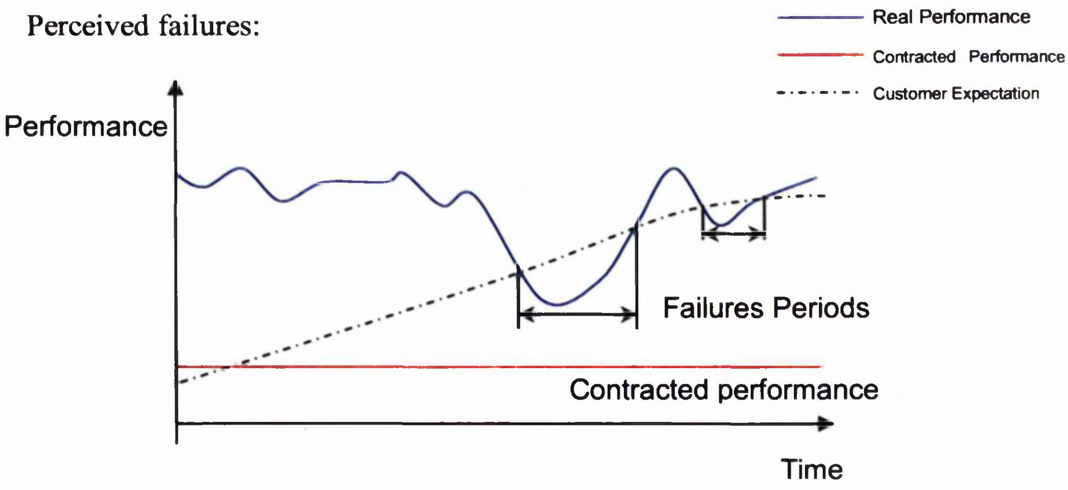


Figure 5: Customer Perceived Failures

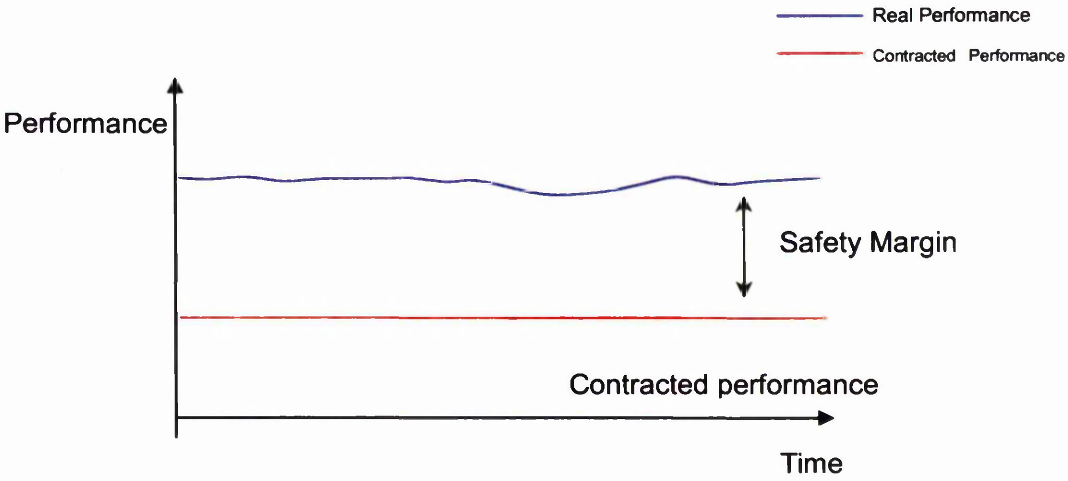


Figure 6: Stable System Performance

3.2.2 Internal Failure and External Failure

Failures of support systems can be divided into two groups: external and internal failures. External failures are those that can be seen by customers and relate to service quality and customer expectations. Typical examples are low quality and late deliverables (contractual failure), and perfunctory and slow reaction to client enquiries (perceived failure)

Internal failures relate to system inefficiency and individual events are not evident to the customer. For example, the transfer of inaccurate data between activities may lead to repeat work and hence increased cost, but the customer may be unaware of the problem if the overall system performance is satisfactory.

3.2.3 Failure State and Operating State

In hardware systems, the failure state of components may often be defined precisely and, if the failure rates of particular components are known accurately, then the overall reliability of a system may be calculated. Redundancy, both active and stand-by, may be included in such calculations.

Such modelling is not generally possible in the case of services. A particular activity can rarely be defined as 'failed'. Rather, it can be seen to slow down, say because there are fewer people working that day, or there may be a delay in the supply of some equipment. It may be that the wrong information (or material) is supplied and that particular activity may need to be repeated.

In a complex service system, individual events such as repeat or tardy work may not cause a failure of the whole system in the sense that the total system performance does not fall below requirements. However, if there is an unfortunate combination of adverse individual events, then a system failure event may be created, which is an 'External Failure'. This may be evident by not achieving KPIs such as response time to an incident or the delivery of a remanufactured product on time.

3.2 Reliability modelling

Although reliability modelling is not a principal theme of this paper, the above discussion of failure characteristics leads naturally to consider how Internal Failures may be modelled to show External Failures (perceived or contractual) in a service support system model.

The system performance is measured by KPIs. An example of a KPI is the time taken to respond to an unplanned fault on a production machine and restore the equipment to working condition. The functional performance of the service support system (with respect to a KPI) is the functional reliability of the system, i.e. *the probability that the system will not perform below a prescribed level, for a defined period of time, with specified resources and operating in a given environment.*

It is possible to create a model of a service system and carry out a simple simulation using MatLab (Stateflow is useful) to give the system reliability. The principal functions (figure 2) are modelled by particular activities which are connected by their respective inputs and outputs in a system network (figure 3). Each activity is modelled by:

- the time taken to carry out the activity (mean and std. dev.)
- the resources required (person-hours)
- the 'reliability' of the activity.

The reliability ($R_{activity}$) of an activity is a measure of the ability of the activity to carry out its function satisfactorily. If, say $R_{activity}$=0.9, then over a long time 10% of the output of this activity is not satisfactory and, in these cases, the activity must be repeated. A simple method is to generate a random number (N_{rnd}) between 0 and 1 at the end of each activity. Using the case of $R_{activity}$= 0.9, if $N_{rnd} > 0.9$ then the activity is repeated. In more complex models, incorrect information may be acted on by subsequent activities and a whole set of activities repeated when the error is discovered at a later time. The failure of the activity (Internal Failure) does not necessarily cause the system to fail, it only necessitates a repeat action which increases the time taken and the resources used to complete the system requirements satisfactorily. However, a number of activity failures can combine to create an External Failure. The overall system reliability is the % of time that the overall performance falls below the contracted level during a defined time period ('failure period/total time' in figure 4). Such a simulation may be run using different resources (and hence cost) and different reliabilities for each activity in order to explore the sensitivity of the system to the performance of particular activities. Thus, system activities may thus be resourced to give, as near as possible, the total system performance characteristic shown in figure 6.

4. Summary

In the general service sector, service failures have been well documented and are often considered as measures of service quality. Contract failures, e.g. delayed flights, account for the majority of failures but customer perceived failure is important. Recovery strategies in the event of a failure are particularly significant, because recovery actions tend to be cost effective and, if good, they can leave a very favourable impression in the mind of their customer. However, the recovery of engineering support systems involving costly hardware is less cost effective due to the financial penalties in contracts. Also, in engineering systems, certain failures cannot be tolerated for safety reasons.

Reliability is a key factor in the performance of service support systems that are part of a Functional (or Total Care) Product because the aim is to provide a guaranteed level of performance over a long period of time. Contract and customer-perceived failures are important. Contract failures are when the system performance with respect to defined KPIs falls below specified levels. Customer-perceived failure can occur if performance falls below expectations even though contracted performance has been achieved. For example, if a high level of performance has been routinely given then that may well become the reference rather than the contracted KPI.

Service support system failures can be external or internal. External failures are those seen by the customer. They occur when a number of individual service activities are poor and their combined effect reduces the overall system performance below contract, or perceived, levels. Internal failures are not visible to the client. They increase the time taken to provide the service, and they increase cost, but they may not have a total effect on the system KPI such that it falls below the contract (or customer-perceived) level.

The functional reliability of a service system may be investigated using a system network model constructed from the activity models. Each activity may be simply modelled using the time taken for the activity, the resources used each time the activity is carried out and the reliability of the activity. The sensitivity of the system to particular activities can be investigated to allocate resources to achieve a stable service provision at minimum cost.

References:

Alonso-Rasgado, T, Thompson, G and Elfström, B-O (2004), 'The Design of Functional (Total Care) Products', *Journal of Engineering Design* 15 (6), pp. 515-540

Bitner M.J., Booms B.H. and Tetreault M.S., (1990), 'The Service Encounter: Diagnosing Favorable and Unfavorable Incidents', *Journal of Marketing*, vol 54, no January, (1990), pp 71-84

Chung B. and Hoffman K.D., (1998), 'Critical Incidents. Service Failures That Matter Most', *Cornell Hotel and Restaurant Administration Quarterly*, vol 39, no June, (1998), pp 66-71

Colgate M. & Norris M., (2001), 'Developing a Comprehensive Picture of Service Failure', *International Journal of Service Industry Management*, Vol. 12 No. 3, 2001, pp. 215-233

Edvardsson, B(1992), 'Service Breakdowns: A Study of Critical Incidents in an Airline', *International Journal of Service Industry Management*, Vol. 3 No. 4, 1992, pp. 17-29

Flanagan J.C., (1954), 'The Critical-Incident Technique', *Psychological Bulletin*, Vol. 51 (1954), pp.327-358Oppenheimer D. and Patterson D. A., (2002), 'Studying and Using Failure Data from Large-Scale Internet Services'

Forbes L.P., Kelley S.W. and Hoffman K.D., (2005), 'Typologies of E-Commerce Retail Failures and Recovery Strategies', *Journal of Services Marketing*, 19/5, (2005) 280-292

Gronroos, C (1998) Marketing services: the case of a missing product. Journal of Business and Marketing. Vol 13, No 4/5, pp322-338

Hill A.V. Collier, D.A., Froehle, C.M., Goodale, J.C., Metters, R.D. and Verma, R. (2002), 'Research opportunities in service process design', Journal of Operations Management 20 (2002) 189–202

Kelley S.W., Hoffman K.D. and Davis M.A., (1993), 'A Typology of Retail Failures and Recoveries', *Journal of Retailing*, vol 69, no Winter, (1993), pp 429-452

Meuter M.L., Ostrom A.L., Roundtree R.I., & Bitner M.J., (2000), 'Self-Service Technology: Understanding Customer Satisfaction with Technology-Based service Encounters', *Journal of Marketing*, Vol.64, July 2000, pp. 50-64

Persson, F., (2004), Service Concept Design, Licenciate Thesis, Lulea University of Technology, Lulea, Sweden

Shostack G.L., (1984), Designing Services that Deliver, *Harvard Business Review*, January-February 1984

Stefan M., (2003), 'Perceived Justice in Service Failure and Recovery Incidents: A Multiple Correspondence', Approach AMA Winter Educators' Conference 2003