TOWARDS A RESILIENT NETWORKED SERVICE SYSTEM

A Thesis Submitted to the College of Graduate Studies and Research

In partial fulfillment of the requirements

For the Degree of Doctoral of Philosophy

In the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon, Canada

By

Junwei Wang

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Mechanical Engineering

University of Saskatchewan

Saskatoon, Saskatchewan (S7N 5A9)

ABSTRACT

Large service systems today are of highly network structures. In this thesis, these large service systems are called networked service systems. The network nature of these systems has no doubt brought mass customized services but has also created challenges in the management of their safety. The safety of service systems is an important issue due to their critical influences on the functioning of society. Traditional safety engineering methods focus on maintaining service systems in a safe state, in particular aiming to maintain systems to be reliable and robust. However, resilience cannot be absent from safety out of many recent disasters that occur in society.

The goal of this thesis is to improve the resilience of networked service systems. Four major works have been performed to achieve this goal. First, a unified definition of service systems was proposed and its relationship to other system concepts was unfolded. Upon the new definition, a domain model of service systems was established by a FCBPSS framework, followed by developing a computational model. Second, a definition of resilience for service systems was proposed, based on which the relationship among three safety properties (i.e., reliability, robustness and resilience) was clarified, followed by developing a framework for resilience analysis. Third, a methodology of resilience measurement for service systems was proposed by four measurement axioms along with corresponding mathematical models. The methodology focused on the potential ability of a service system to create optimal rebalancing solutions. Two typical service systems, transportation system and enterprise information system, were employed to validate the methodology. Fourth, a methodology of enhancing resilience for

service systems was proposed by integrating three types of reconfigurations of systems, namely design, planning and management, along with the corresponding mathematical model. This methodology was validated by an example of transportation system.

Several conclusions can be drawn from the work above: (1) a service system has a unique characteristic that it meets humans' demand directly, and its safety relies on the balance between the supplies and demands; (2) different from reliability and robustness, the resilience of a service system focuses on the rebalancing ability from imbalanced situations; (3) it makes sense to measure the resilience of a service system only for a particular imbalanced situation and based on evaluation of rebalancing solutions; and (4) integration of design, planning and management is an effective approach for improvement of the resilience for a service system.

The contributions of this thesis can be summarized. Scientifically, this thesis work has improved our understanding of service systems and their resilience property; furthermore, this work has advanced the state of knowledge of safety science in particular having successfully responded to two questions: is a service system safe and how to make a service system safer? Technologically or methodologically, the work has advanced the knowledge for modeling and optimization of networked service systems in particular with multiple layer models along with the algorithms for integrated decision making on design, planning, and management.

ACKNOWLEDGEMENTS

I would like to express my great thank to Professor W.J. (Chris) Zhang for his excellent guidance during this thesis work. The depth and width of his knowledge is admirable. His enthusiasm to the scientific research always encourages me.

I would also like to express my sincere gratitude to all my committee members, Prof. Greg J. Schoenau, Prof. Ralph Deters and Prof. FangXiang Wu, for their valuable suggestions and comments during my committee meetings.

I gratefully acknowledge the financial support from Natural Science and Engineering Research (NSERC) through the scholarship.

I would like to thank all my families for their continuous support and love.

DEDICATED TO

My Parents in China

My Wife Junfeng

My Sons Arron and Alex

ACRONYMS

AEB	Accident Evolution and Barrier
AET	Average Evacuation Time
CREAM	Cognitive Reliability and Error Analysis Method
DOE	Design of Experiments
EDD	Earliest Due-Date
ETA	Event Tree Analysis
FBS	Function-Beauvoir-State
FCBPSS	Function Context Behavior Principle Structure State
FMEA	Fault Modes and Effects Analysis
FP	Flow Pattern
FRAM	Functional Resonance Accident Model
FTA	Fault Tree Analysis
GA	Genetic Algorithm
GD	Good-Dominant
HRO	High Reliability Organization
IA	Iterative Algorithm
IS	Infrastructure System
MCF	Minimum Cost Flow
MCFA	Minimum Cost Flow Algorithm
MMS	Man-Machine System
MNE	Maximum Number of Evacuees
NAT	Normal Accident Theory
OASIS	Organization for the Advancement of Structure Information Standards

OO	Objective-Oriented
P-MCMF	Priority-based Minimum Cost Multicommodity Flow
PSO	Particle Swarm Optimization
PSS	Product-Service System
RS	Recovery Schedule
S-D	Service-Dominant
SDLC	Systems Development Life Cycle
SNR	Signal-to-Noise Ratio
SO	Service-Oriented
SOA	Service-Oriented Architecture
SPT	Shortest Processing Time
SRA	Safety Resource Allocation
SS	Substance System
STAMP	System-Theoretic Accident Model and Processes
TEN	Time Expanded Network
TET	Total Evacuation Time
TI	Transportation Infrastructure
TPSO	Three-Pattern Particle Swarm Optimization
TS	Transportation Substance
UML	Unified Modeling Language
W3C	World Wide Web Consortium

Table of Contents

List of Tables

List of Figures

CHAPTER 1
INTRODUCTION

1.1 Background and Motivation

Modern human society critically relies on different service systems which provide timely and sustainable services to our daily lives and social production activities. One of the most important features of modern service systems is that they work in a networked manner. A generic network includes a set of nodes which are connected to each other in some ways. The nodes not only represent humans but also technical systems. The networked service systems are thus complex socio-tech systems.

Unfortunately, the safety of networked service systems is still a great concern. A few recent service system failures have supported this concern. The successive breakdown of the power supply system in the Northern India happened in July 2012, causing one-half of the country into serious trouble, and this breakdown further caused failures of other critical service systems, such as transportation system, financial system, water supply system, and hospital system [India today online 2012]. In 2003, four developed counties (United States [CNN 2003], Canada [CNN 2003], United Kingdom [BBC 2003b] and Italy [BBC 2003a]) all experienced serious blackout issues. Another example is the recent accident of communication system in Calgary, Canada; a communication system provided by Shaw, the largest local telecommunication provider, failed due to an explosion in July 2012 [Metronews 2012].

Indeed, according to [Zhang 2008], the network structure of service systems makes them a higher potential of cascading failures than non-network structures. In the network structure, several critical service systems become more interdependent on one another, and therefore, the cascading effect more likely takes effect. The blackout in the Northern India did cause dysfunctions of transportation and bank systems. The failure of the Calgary communication system has shown that the service system is more sensitive to internal or external disturbances. It is the motivation of this thesis to study the safety of a highly networked service system – in particular how to design and operate a network service system so that it can be much safer.

Unfortunately, a preliminary study revealed that two important concepts, (1) service and service system and (2) safety and safety engineering have not been clearly defined in the current literature. Therefore, there is a need to clarify these two concepts. At the outset of this research, seeking the answer to the two aforementioned questions has been attempted.

It is well known that there are three important safety properties of a system, namely reliability, robustness and resilience and there are three corresponding research fields, reliability engineering, robustness engineering and resilience engineering. Reliability engineering and robustness engineering are relatively well developed; reliability focuses on different accident models and robustness focuses on how to maintain a system to function to meet the requirement. Resilience is a popular concept in material science [Nagdi 1993], psychology [Luthar et al. 2000] and ecology [Gunderson 2000]. Recently, researchers from system safety engineering have paid attention to the concept of resilience engineering by introducing the resilience concept into safety engineering [Hollnagel et al. 2006]. From an engineering perspective, resilience has been viewed as a property of a system regarding its ability to recover its basic [or generic] function after the system is attacked [or disturbed] to such an extent that either system is physically broken or beyond its yield state [adopted from Zhang 2007].

Indeed, the system failure examples as previously mentioned have indicated that the current methodologies of reliability engineering and robustness engineering are not adequate to having a safe system with a tolerable disruption. This implies that a new effort on resilience engineering is worthwhile to be taken.

The overall objective of the research in this thesis was to improve the understanding of resilience of a networked service system and develop technology to improve its resilience. A preliminary study by the author led to the view that to achieve this overall objective, two issues need to be addressed, namely (1) how to measure the resilience of a networked service system and (2) how to design and operate a networked system for a controlled resilience.

Regarding the first issue, resilience can be investigated from a perspective of the safety performance of systems under abnormal situations. This calls for a deep understanding of the service system, which was in fact missing at the time this research was initiated. Regarding the second issue, it is necessary to provide a measure for the resilience of a service system.

1.2 Objectives

To achieve the overall objective of the research, the following specific objectives were defined.

Objective 1: to have a clear understanding of the networked service system and to build a domain model to lay down a foundation for measure, design and operate a service system for high resilience and other performances.

Objective 2: to have a clear understanding of resilience of the service system; in particular, to build the intricate relationships among reliability, robustness, resilience, and safety.

Objective 3: to develop a methodology to measure the resilience of a networked service system.

Objective 4: to develop a methodology for improving the resilience of a networked service system.

1.3 Organization of the Thesis

The remainder of this thesis is laid out into seven chapters as follows.

Chapter 2 presents a literature review on the service system and resilience engineering. The review will further confirm the validity of the proposed objectives for this research. Indeed, this review will come to the conclusions including (1) a unified definition of service and service system has been absent and (2) a relationship among reliability, robustness, resilience and safety has been absent.

Chapter 3 presents a unified definition of service and service system, which lays down a foundation to discuss the relationship among reliability, robustness, resilience and safety. The discussion in this chapter has further made it possible to discuss the measure and measurement of resilience for a networked service system, design and operation of a networked service system for higher resilience with consideration of cost and time.

Chapter 4 provides a domain model and an analysis of service system, with which the variables to affect resilience of a service system are derived from the domain model.

Chapter 5 presents a clear understanding of resilience for the service system by (1) proposing new definitions of safety and resilience, (2) clarifying the relationships among reliability, robustness, and resilience, and (3) developing a framework for resilience analysis.

Chapter 6 discusses measurement axioms and corresponding mathematical models for resilience of a networked service system. Therefore, given a particular networked service system, its resilience can be measured quantitatively.

Chapter 7 presents a methodology for the resilience improvement by (1) building an integrated strategy of design, planning and management by a framework and a general mathematical model, and (2) developing a detailed mathematical model with its corresponding algorithms to conduct the integrated strategy in a validation case of transportation system.

Chapter 8 gives the conclusions and future work.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

There are two important concepts and related topics in this thesis: service system and resilience engineering. This chapter first gives a literature review on the concept of service system as well as a similar concept called service-oriented system and discusses the relationship between these two concepts in Section 2.2. There are two important perspectives to investigate the resilience concept and resilience engineering, as shown in Figure 2.1. The first perspective is to understand the resilience concept through examination of its origin and definition in different disciplines. This perspective is helpful for understanding of connation of resilience concept and its basic features. The second perspective is to understand resilience engineering as a branch of safety engineering. The two perspectives are discussed in Section 2.3 and 2.4, respectively. As last, a conclusion is given in Section 2.5.

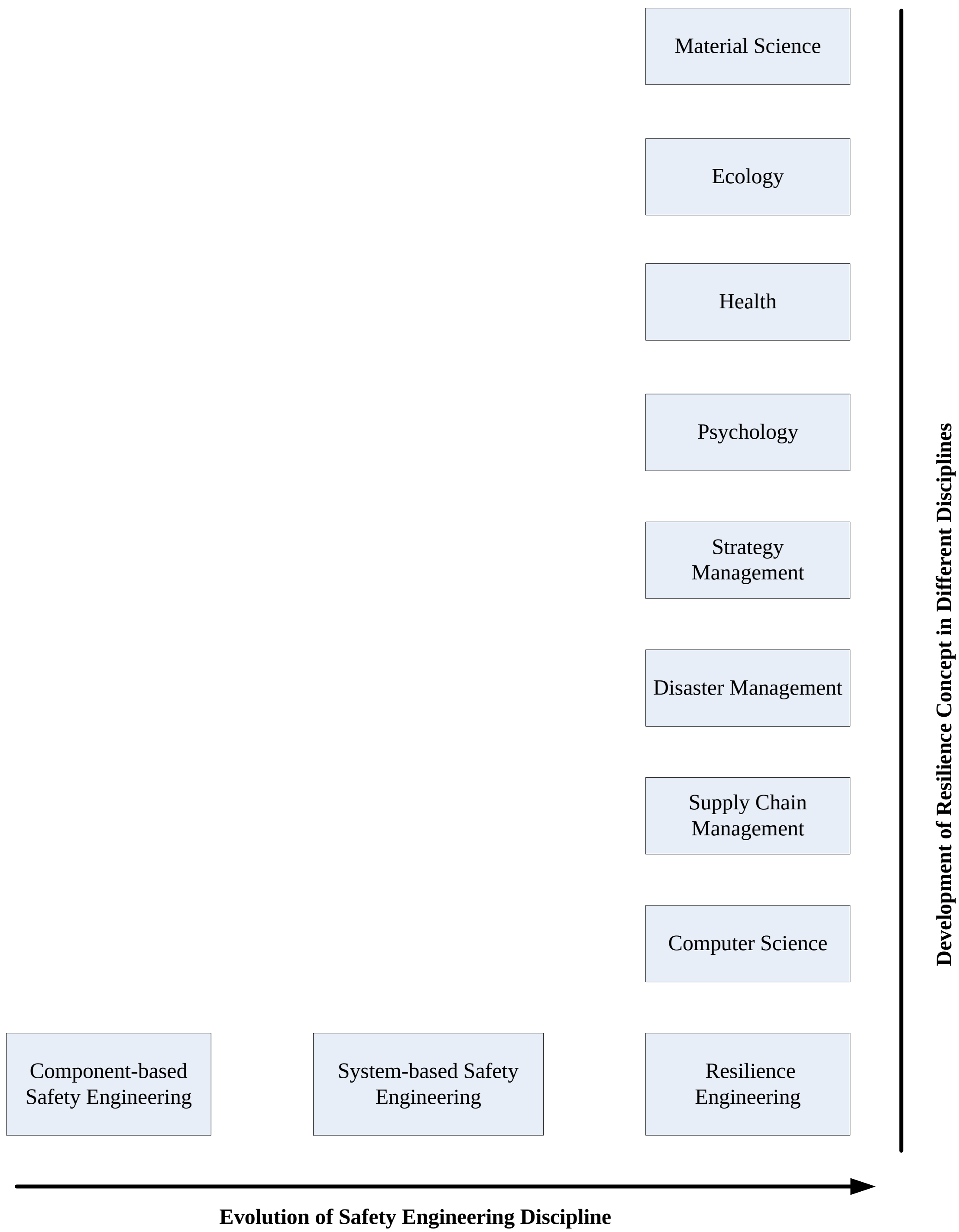

Figure 2.1 Two perspectives for understanding of resilience and resilience engineering

2.2 Service System and Related Concept

2.2.1 Service System

2.2.1.1 Origin of service system

The concept of the service system was proposed from the economic or value point of view. That is, from a point of view of economy, a system can broadly be classified into three categories: agricultural systems [Rigby and Cáceres 2001], manufacturing systems [Smith 2003] and service systems [Spohrer et al. 2007], which are further relevant to agricultural economy, manufacturing economy and service economy, respectively. Over the last two decades, emphasis on manufacturing economy has been shifted to service economy [Krishnamurthy 2007]. According to the report by the International Labour Organisation in 2007, such a shift was especially true in the developed nations and the total number of jobs in the service economy is more than that in the agricultural and manufacturing economy [IfM&IBM 2008]. These significant changes in the field of economy have greatly raised researcher’s attention to the service system worldwide.

The earliest appearance of the term 'service system' is perhaps in a book entitled 'stochastic service system' [Riordan 1962]. The emphasis of this book is, however, on use of a mathematical approach to study telephone systems, especially the telephone traffic congestion problem. Obviously, despite the name 'service' used in this book, there is no sense of the service economy at that time. The earliest research on service system (which has the same concept as the one herein) was conducted by Levitt [1972] and Mills and

Mober [1982]. They attempted to apply the manufacturing concept to service. Their motivation and idea can be well understood from a point of view of economy.

2.2.1.2 Concepts: service and service system

(1) Definition of service

Several important definitions of service proposed in literature are listed in Table 2.1.

Table 2.1 Definitions of service

	Definition	Literature
1	"Service is a time-perishable, intangible experience performed for a customer acting in the role of co-producer."	Fitzsimmons and Fitzsimmons 2004
2	"Service is the application of competences by one entity for the benefit of another."	Vargo and Lusch 2004, 2006
3	"Service is a provider/client interaction that creates and captures value."	IBM 2011b
5	"Service refers to economic activities offered by one party to another."	Lovelock et al. 2009
6	"Service is a process by which the provider fulfills a mission for a client so that value is created for each of the two stakeholders."	Lau et al. 2011

These definitions of service have captured some important features of service. However, they are too general to be useful; in particular there is a difficulty with them to distinguish a service from a product, manufacture, and agriculture. In fact, following these definitions, agriculture and manufacture, along with products and goods, are both services [Lau et al. 2011]. These definitions may further lead to the concept that "all economies are service economy", as pointed out by Vargo et al. [2008]. This problem (i.e., all inclusive) is inherited to the concept of service system, which will be discussed later in this chapter.

The shift of economies to service economy, as mentioned above, may have some sense of business benefit. For instance, IBM seems to lead the research of service science and tends to sell their business solutions to all the firms and sectors. Therefore, a more general definition of economy is helpful for their business. This should not however compromise the need to give each concept a distinct definition so that research can be built upon a common framework.

(2) Definition of service system

The 'service system' has been given different definitions from different points of view in literature. The existing definitions of the service system in literature are listed in Table 2.2, where the right column ‘category’ is designed by this thesis (i.e., these definitions are further put into three categories).

The first category (definition 1 and definition 2) in Table 2.2 has a distinct feature that a human-in-the loop is considered as a key feature in service system. In particular, definition 1 considers the service as a process driven by the customer. This definition can however bring a manufacturing system and a product system into the domain of service system. If one fixes on the definition of customer as a human individual, a cell phone which is a product system then becomes a service system, as in this case, a user is a customer, and the user’s input on the cell phone triggers the work of the phone. Moreover, definition 2 cannot distinguish a service system from a manufacturing system. Broadly, an organization which has a manufacturing business is considered to be a manufacturing system. Therefore, according to definition 2, Boeing appears to be a manufacturing system. However, Boeing also has a service business, and Boeing can then be considered to be a service system. In short, bringing humans in the system is not enough to give an identity of service system.

Table 2.2 Definitions of service system

	Definition	Literature	Category
1	"With a service process, the customer provides significant inputs into the production process."	Sampson and Froehle 2006	I
2	"A service system is a voluntary and human usable system, that is, a usable system which contains a significant level of people or organizations as components during operation and needs voluntary engagement of an external person/organization to produce value."	Pinhanez 2009	I
3	"A service system is a work system that produces services."	Alter 2003, 2006, 2008b	II
4	"A *service system* is defined as a value-coproduction configuration of people, technology, other internal and external service systems, and shared information (such as language, processes, metrics, prices, policies, and laws)."	Spohrer et al. 2007, 2008	III
5	"A service system is defined as a dynamic configuration of resources (people, technology, organisations and shared information) that creates and delivers value between the provider and the customer through service."	IfM & IBM 2008	III
6	"A service system is a composite of agents, technology, environment, and/or organization units of agents and/or technology, functioning in space-time and cyberspace for a given period of time."	Stanicek and Winkler 2010	III

The second category (definition 3) has a distinct feature that the service system and manufacturing system have the same structure but are capable of producing either a product or service. The so-called work system follows the definition given by Alter [2003], which says: “a work system is a system in which human participants and/or machines perform work using information, technology, and other resources to produce products and services for internal or external customers.”

The third category (definition 4, 5 and 6) has a distinct view that a service system is a complementary component of the economic exchange [Demirkan et al. 2011]. There are a couple of concerns with definition 4. First, knowledge is not included. It is noted that knowledge and information are of distinct concepts; see [Zhang 1994]. Therefore, knowledge should be included in the definition of service system alongside with information. It is noted that Vargo et al. [2008] has also realized this issue and added knowledge into the definition of service. Second, this definition is all-inclusive, as stated by the authors, “individuals, families, firms, nations, and economies are all instances of the service system.” As pointed out before, the writer considered that the all-inclusive approach can hardly differentiate any individual system, which is not workable in practice. Third, this definition seems to put emphasis on value and value co-production or co-creation as a feature of service system, which is equally impossible to differentiate different economies. In fact, following this definition, one can also arrive at the point that all economies are service economies [Vargo et al. 2008]. Definition 5 emphasizes the exchange between different economic resources and economic values according to [IfM & IBM 2008]. However, this definition is also all-inclusive, ranging from individual people to businesses, nations and even ecosystems. Definition 6 is an extension of definition 4 by including the context and time to give a sense of dynamics.

Besides the definitions above, there are also definitions of service system from a point of view of the nature of various services. For instance, Lusch and Vargo [2006] defined service as: "service may refer to a kind of action, performance, or promise that is exchanged for value." Krishnamurthy [2007] outlined four features of a service as: (1)

intangible, (2) consumed at the time it is produced, (3) provision of value-adding in different forms, and most importantly, (4) co-production. Regarding the last feature, Tien and Berg [2003] explained: "co-production means that the consumer and provider are communicating constantly, re-evaluating the need of the customer and the manner in which the customer is being satisfied."

The above definitions have difficulties to distinguish a service system from other systems, such as agricultural systems, manufacturing systems, and product systems. For instance, in the agricultural system, humans and technologies are also included. Modern agricultural systems are highly automated similar to manufacturing systems. Emphasis on technology, people, and organization for the manufacturing system can be dated back as early as to the 1990s; see [Zhang et al. 1997]. Further, the first and second features of service as described by Krishnamurthy [2007] fail to include the transportation system which passes goods from place A to place B. The nature of co-production is the customer participation in businesses, which is not only for service but also for manufacturing according to Li et al. [2004]. It appears that an identity of the service system is still not clear from the current literature.

2.2.1.3 Related work

The service system has evolved into a new discipline today and has been studied from different perspectives: science, technology, engineering and management; furthermore, there are different names or labels which have something to do with service system, e.g., service science [Spohrer and Maglio 2008], service engineering [Bullinger et al. 2003 and

Mandelbaum 1998], service science, management and engineering (SSME) [Maglio et al. 2006], and service systems engineering [Tien and Berg 2003]. It is noted that SSME was proposed by an IBM research team and further simplified as service science; in particular, service science is viewed as a study of service systems from different perspectives, including science, engineering and management. Next, a summary of the related work is given from these perspectives.

(1) From a science perspective, researchers have investigated the origin of service and service system and studied important features of service, service system and service science discipline, based on which the definitions of these important concepts are proposed. Some related work has already been introduced in Section 2.2.1.1 and Section 2.2.1.2. One important work is the service-dominant (S-D) logic proposed by Vargo and Lusch [2004, 2006], which is considered as an economic foundation of the service system, instead of the traditional good-dominant (G-D) logic. The difference between the S-D logic and the G-D logic has been summarized in the work of [Constantin and Lusch 1994, Vargo and Lusch 2004] as: "S-D logic focuses on the action of operant resources (those that act upon other resources), such as knowledge and skills, whereas G-D logic focuses on the exchange of operand resources (those that an act or operation is performed on, such as goods)." It is noted that, similar with the problems in the definitions of service and service system discussed in Section 2.2.1.1 and 2.2.1.2, S-D logic is also an all-inclusive concept which considers all the economies as service economy.

(2) From an engineering perspective, researchers have focused on how to develop and improve the performance of a service system, including improving efficiency, effectiveness, sustainability and safety [Bullinger et al. 2003, Spohrer et al. 2007]. The methodologies of modeling [Stanicek and Winkler 2010], design [Shimomura and Arai 2004, Karni and Kaner 2006, Glushko and Tabas 2009], measurement [Wang et al. 2010a, Xie 2011] and control [Mascio 2003] were proposed. It is noted that in the current literature, a service system was only examined from a particular phase of the whole system life-cycle; in particular, a service system was investigated from the phase of design, planning and control separately [Wang 2010].

(3) From a management perspective, the issues in operations management, marketing, human resources have been widely discussed [Rai and Sambamurthy 2006, Sengupta 2006, Li et al. 2007]. The most important work was given by Daskin [2010] in his recent book, service science. Although his book was named as service science, he has investigated the service system from the management point of view only; in particular, he has provided comprehensive discussions on quantitative methodologies, particularly optimization and queuing theory, has applied them to solve a variety of problems in service systems, and has aimed at facilitating decision-making during their operations and management.

2.2.2 Service-oriented System

2.2.2.1 Introduction

Different from service system, the concept of service-oriented system is proposed from a technological point of view. From a technological perspective, a service-oriented system (SO system) is a system based on service-oriented architecture (SOA). SOA is a way of designing software systems; in particular, it aims at (1) offering a flexible infrastructure and operation environment and (2) generating the independent and reusable functions (or services) to end-user applications or other services in a network via standard interfaces, according to Papazoglou et al. [2007]. In the context of SOA, a service focuses on the autonomous means of retrieving and processing information according to Liu and Deters [2008], which is different from the definition discussed in Section 2.2. The details of the concepts are discussed in the next section.

2.2.2.2 Concepts: service and service-oriented architecture

The definition of service-oriented system depends on the definitions of service and SOA, respectively. There are different definitions of service and SOA, among which the definitions proposed by three important organizations, W3C (World Wide Web Consortium), OASIS (Organization for the Advancement of Structured Information Standards) and the open group, are widely accepted, as listed in Table 2.3.

Table 2.3 Definitions of service and SOA

	Service	SOA	
W3C	A service is a set of actions that form a coherent whole from the point of view of service providers and service requesters. A service requester is the entity that is responsible for requesting a service from a service provider; in particular, service requester is a web service agent. A Service Provider is an agent that is capable of and empowered to perform the actions associated with a service; in particular, service provider is a web service agent.	A Service Oriented Architecture (SOA) is a form of distributed systems architecture that is typically characterized by the following properties: Logical view, Message orientation, Description orientation, Granularity, Network orientation and Platform neutral.	[Booth et al. 2004]
OASIS	A service is a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.	Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.	[OASIS 2006]
The open group	A service: (1) Is a logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports) (2) Is self-contained (3) May be composed of other services (4) Is a “black box” to consumers of the service	Service-Oriented Architecture (SOA) is an architectural style that supports service orientation. Service orientation is a way of thinking in terms of services and service-based development and the outcomes of services. An architectural style is the combination of distinctive features in which architecture is performed or expressed.	[the open group 2011]

The definitions proposed by W3C are from a web service perspective; in particular, service refers to the web service with the form of software. The definitions proposed by OASIS and the open group are more general; in particular, they are from a business process perspective [IBM 2011a]. It is noted that none of these definitions refers to any specific architecture.

Box [2004] summarized the principles of SOA into four features: (1) Boundaries are explicit, (2) Services are autonomous, (3) Services share schema and contract, not class, and (4) Service compatibility is determined based on policy. A common goal of the four features is to develop loosely coupled distributed systems based on the notion of services. Liu and Deters [2008] concluded six characteristics for the service-oriented systems: (1) distributed, (2) homogeneous and yet heterogeneous, (3) open, (4) dynamic, (5) hierarchical and heterarchical, and (6) scalable.

2.2.2.3 Related work

The service-oriented system is mainly examined from a perspective of engineering; however, some researchers studied the service-oriented system from perspectives of science and management as well. Therefore, similar to service system, related work is summarized from three perspectives: science, engineering and management.

(1) From a science perspective, the basic concepts, such as service, software as a service, service-oriented computing, service-oriented architecture, and service-oriented system were discussed in [Turner et al. 2003, Jones 2005, Papazoglou et al. 2007]. Furthermore, a discipline, called service-oriented science was proposed in [Foster 2005].

(2) From an engineering perspective, researchers have investigated main aspects of the service-oriented system life cycle [Kontogiannis et al. 2007], such as collaborative specification and modeling [Karle and Oberweis 2008], design [Papazoglou and Heuvel 2006], implementation [Schmidt et al. 2005, Jameela et al. 2011], evaluation [Tsai 2005], operation and maintenance and different applications [Lane and Richardson 2011]. It is noted that most of the studies only addressed particular areas of life-cycle; of the studies that attempt to address an entire life-cycle, few of them have been validated with real-life scenarios [Lane and Richardson 2011].

(3) From a management perspective, researchers have addressed some important issues, such as service level management [Muller 1999, Tosic et al. 2005], management information modeling and communication [DMTF 2005], different management techniques [Almeida et al. 2006, Tseng and Wu 2007], and self-management [Cao et al. 2004, Lewis et al. 2004]. Liu and Deters [2008] have given a good literature review on the topics above and have summarized the characteristics of service-oriented systems as well as the major challenges of management. From a broader management point of view, there are some work on the business level of service-oriented systems [Cherbakov et al. 2005, Demirkan et al. 2005, Demirkan et al. 2008].

2.2.3 Relationship between Service System and Service-oriented System

(1) The concepts of service system and service-oriented system are proposed from different perspectives. As discussed in Section 2.2.1 and Section 2.2.2, the service system

concept was proposed from an economic perspective while the service-oriented system concept was proposed from a technological perspective.

(2) The two concepts may be coupled. In real-world applications, the two concepts may be coupled. On one hand, a service system may be built up based on different technologies; in particular, it may be built up based on the service-oriented architecture technology, which leads a service system to be a service-oriented system. Therefore, the concepts of service-oriented service system and non-service-oriented service system make sense. On the other hand, the service-oriented architecture may be applied in different application domains, including manufacturing systems, service systems and so on. Therefore, the concepts of service-oriented manufacturing system and service-oriented service system make sense.

For example, Ordanini and Pasini [2008] discussed that the service-oriented architecture is a good tool to implement the service co-production and value co-creation according to the service-dominant (S-D) logic. In Section 2.2.1, it has been discussed that service co-production and value co-creation are two important features of a service system and S-D logic is the economic foundation of service system. IBM also has the same point of view, as shown in Figure 2.2. They consider that the service-oriented architecture is with the technology domain, and different application systems built upon SOA are with the application domain. It is noted that in Figure 2.2, communication system, e-commerce system and finance system are all service systems.

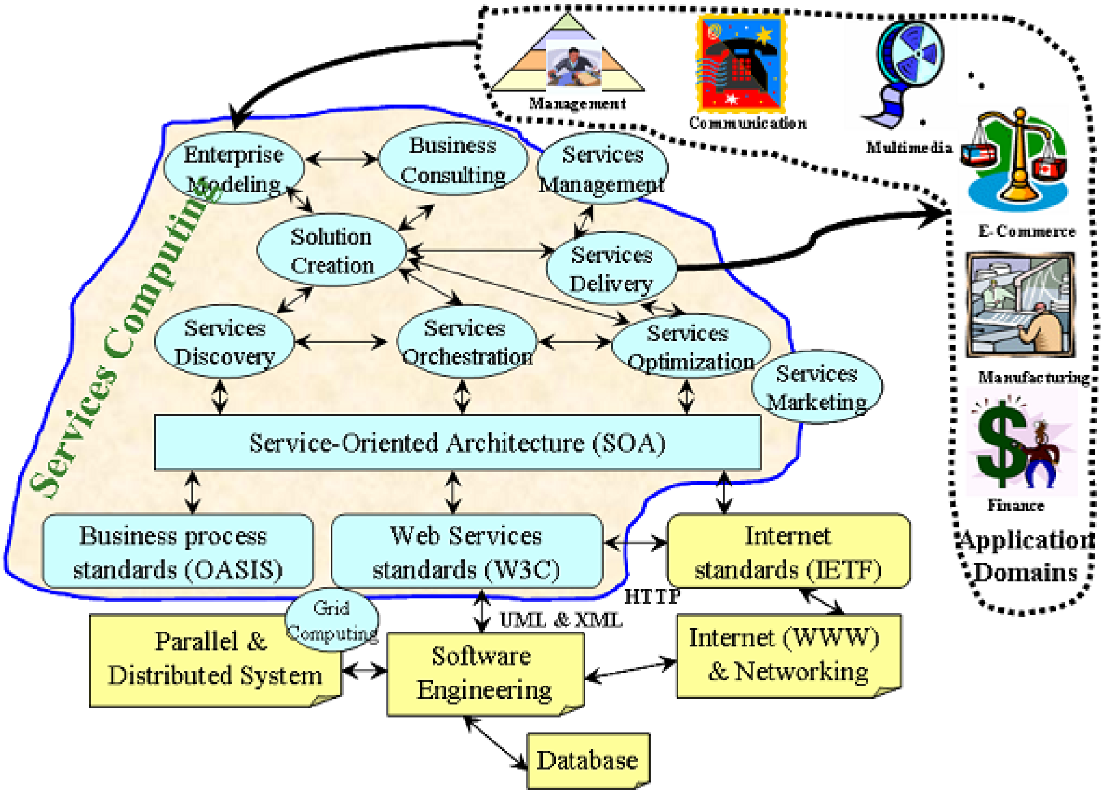

Figure 2.2. Service-oriented architecture with different application domains (Adopted from: https://researcher.ibm.com/researcher/view_pic.php?id=152)

(3) Both concepts can be examined from a system's point of view. Although the two concepts are proposed from different perspectives, they can be examined from the engineering perspective; in particular, both of them can be examined from a system's point of view; furthermore, the research methodology on one concept may be applied to the other.

In the current literature, a service system was only examined from a particular phase of the whole system life-cycle [Wang 2010]; in particular, a service system was investigated from the phase of design, planning and control separately. The literature review in Section 2.2.2 has shown that the same problem remains for the service-oriented system.

In the information system management, there is a particular concept to describe the whole life-cycle of the systems, called systems development life-cycle. The systems development life cycle (SDLC) is a conceptual model used in systems engineering, information system and software engineering, which describes the stages involved in a system development project, from an initial feasibility study through maintenance of the completed application. From a system's point of view, both service system and service-oriented system have subsystems of infrastructure, substance and management and they can be studied with a similar system approach.

2.3 Resilience Concept: Its Origin and Definitions in Different Disciplines

This section first introduces the origin of the resilience concept, particularly the origin of the resilience term in English. As stated in Section 2.1, resilience has been a multidisciplinary concept. Besides safety engineering, there are many other disciplines that also employ this concept. Thus, Section 2.3.2 introduces the development of the concept of resilience in different disciplines. Section 2.3.3 presents and discusses definitions of the resilience concept in these disciplines (other than safety engineering) as well as the features of the definitions. The resilience concept in the context of safety engineering will be discussed in Section 2.4.

2.3.1 Origin of Resilience Concept

The word *resilience* was introduced into English language in the early 17th century from the Latin word *resilire* with the meaning of leaping back or rebounding [McAslan 2010, Oxford dictionary 2012]. The current definitions of resilience in the dictionaries are listed in Table 2.4.

Table 2.4 Definitions of the term *Resilience* in dictionaries

	Definition	
I	1: the ability of a substance or object to spring back into shape; elasticity 2: the capacity to recover quickly from difficulties; toughness	[Oxford Dictionary 2012]
II	1: the capability of a strained body to recover its size and shape after deformation caused especially by compressive stress 2: an ability to recover from or adjust easily to misfortune or change	[Merriam Webster 2012]
III	1: the power or ability to return to the original form, position, etc., after being bent, compressed, or stretched; elasticity. 2: ability to recover readily from illness, depression, adversity, or the like; buoyancy.	[Dictionary.com 2012]

The definitions above show that the resilience of an object refers to its ability of returning to the original state. This concept of resilience has been introduced into different academic disciplines to express the property of different objects or systems.

2.3.2 Development of the Resilience Concept in Different Disciplines

The first use of resilience concept in the academic work could be classified into material science [McAslan 2010]. In the 19th century, Tredgold used the resilience concept to represent the timber's property of sustaining sudden and severe loads without breaking in two of his papers [1818a, 1818b]. Mallet also used the resilience concept to describe the performance of materials under severe conditions and discuss the suitability of materials in the construction of vessels and buildings [Fairbairn 1865, Mallet 1856, Mallet 1862].

The second area which has employed the resilience concept is ecology. Holling [1973] introduced the concept of resilience into the ecology and defined the resilience as a property of an ecological system which enables the system to absorb changes and still persists. It is noted that Holling's work [Holling 1973] has an important impact on the development of the resilience concept within ecology and many other disciplines. Thus, the details of his work, particularly his two main contributions on the ecological resilience are introduced in the following.

His paper in 1973 was the first attempt on the study of the ecological resilience. He considered two properties of an ecological system, resilience and stability and compared the two properties [Holling 1973]. He defined the stability as "the ability of a system to return to an equilibrium state after a temporary disturbance", as opposed to the resilience defined by himself in 1973 in [Holling 1973]. He further stated in [Holling 1973]: “a system may be of high resilience but has low stability, as it may fluctuate greatly.”

Holling and other researchers summarized key features of ecological systems and discussed the two categories of understandings of resilience concept in the ecological literature, and they further extended the resilience concept to an integrative theory to understand and analyze the complexity, dynamics and sustainability of ecological, economic and social systems in [Holling 2001, Holling and Gunderson 2002, Holling et al. 2002]. Their theory was also referred to as resilience theory [Redman and Kinzig 2003, Redman 2005]. Holling summarized the key features of ecological system into four points [Holling 1996, Holling and Gunderson 2002]: "(1) change is neither continuous and gradual nor consistently chaotic, (2) Spatial attributes are neither uniform nor scale invariant over all scales, (3) ecological systems do not have a single equilibrium with homeostatic control to remain near it, and (4) policies and managements that apply fixed rules for achieving constant yield, independent of scale, lead to systems that increasingly lose resilience." Two definitions of resilience in the ecological literature summarized by Holling are engineering resilience and ecological resilience [Holling 1996, Holling and Gunderson 2002]. Engineering resilience only considers one equilibrium state of the system; while ecological resilience views that an ecological system has multiple equilibrium states. It is noted that the definition of engineering resilience is not the definition of resilience in the context of engineering; rather it is a definition in the context of ecology. In the best knowledge of the writer, Holling favoured the ecological resilience definition.

Another important application of the resilience concept is to health and social work. It is also called resilience theory, which has the same name yet a different context from the resilience theory discussed above in the field of ecology. The resilience theory in the context of health and social work focuses on the resilience of human and human society, including individual resilience, family resilience, community resilience, resilience-based policy and resilience in social work; see the work of [VanBreda 2001].

Besides the early applications in the three disciplines discussed above, the resilience concept has been introduced into many other fields and has become to be a multidisciplinary concept. Different disciplines may focus on different perspectives.

Some efforts have been done on the classification of resilience definitions [VanBreda 2001, Muddada 2010, Bhamra et al. 2011]. Table 2.5 lists the different disciplines that have adopted the resilience concept.

Table 2.5 Different disciplines with application of resilience concept

	Discipline	Literature
1	Material Science	Tredgold 1818a, 1818b, Nagdi 1993
2	Ecology	Holling 1973, Gunderson 2000, Walker et al. 2002
3	Health and social work	VanBreda 2001, Phaneuf 2003, Kaplan et al. 1996
4	Psychology	Luthar et al. 2000, Luthans et al. 2006
5	strategy management	Hamel and Valikangas 2003
6	supply chain management	Christopher and Peck 2004, Sheffi 2005, Muddada 2010
7	computer science	Najjar and Gaudiot 1990, Laprie 2008
8	Disaster management	Bruneau et al. 2003, Paton et al. 2000
9	Safety Engineering	Hollnagel et al. 2006
10	Archaeology	Redman 2005, Redman and Kinzig 2003

2.3.3 Definitions of the Resilience Concept in Different Contexts

In each discipline, the resilience concept has been studied in the context of a particular object or system. Table 2.6 shows definitions of resilience for various objects or systems. Table 2.6 indicates that although the resilience concept has a common feature of springing back to a normal situation after change as the meanings in the dictionaries, these definitions have different focuses. Next, several remarks are given for further explanation.

Table 2.6 Definitions of resilience in the context of different objects

	Context	Definition	Literature
1	Material	Resilience is the ability of the material to return to its original shape after temporary deflection.	Nagdi, 1993
2	Ecological system	Resilience is a measure of the ability of a system to absorb changes of state variables, driving variables, and parameters, and still persist.	Holling 1973
3	Socio-ecological System	Resilience is the potential of a system to remain in a particular configuration and to maintain its feedbacks and functions, and involves the ability of the system to reorganize following disturbance-driven change.	Walker et al. 2002
4	Individual	Resilience is the capacity to maintain competent functioning in the face of major life stressors.	Kaplan et al. 1996
5	Family	Family resilience refers to the characteristics, dimensions, and properties of families which help families to be resistant to disruption in the face of change and adaptive in the face of crisis situations.	McCubbin and McCubbin, 1988
6	Community /Organization	The ability of social units (e.g., organizations, communities) to mitigate hazards, contain the effects of disasters when they occur, and carry out recovery activities in ways that minimize social disruption and mitigate the effects of future earthquakes.	Bruneau et al. 2003
7	Organization /Business /Company	Resilience refers to a capacity for continuous values, processes, and behaviors that reconstruction. It requires innovation with respect to those organizational systematically favor perpetuation over innovation.	Hamel and Valikangas 2003
8	Company	Resilience refers to the ability of a company to bound back from a large disruption-this includes, for instance, the speed with which it returns to normal performance levels (production, services, fill rate, etc.)	Sheffi 2005
9	Supply chain	Resilience is the ability of system to return to its original or desired state after being disturbed.	Christopher and Peck 2004
10	Computer network	A measure of network fault tolerance.	Najjar and Gaudiot 1990

Remark 1: The resilience of a system is defined and understood based on the examination of the system's key features. As discussed in the aforementioned part of this section, Holling's definition of ecological resilience was proposed based on his summary of

ecological system's four key features [Holling1973]. Thus, different features of different systems result in different resilience understandings and then concepts.

Remark 2: For a particular system, the resilience concept may have different definitions which come from different perspectives or disciplines. For example, the individual resilience was examined in both health and psychology; the community resilience was studied in both disaster management and social work. In Table 2.6, definition 6 and definition 7 could be viewed as definitions of organization resilience; in particular, definition 6 was proposed from a perspective of disaster management and definition 7 was proposed from a perspective of business management. Both definition 7 and definition 8 could be viewed as company resilience; in particular, definition 7 was proposed from a perspective of business management which focuses on the company itself and definition 8 was proposed from a perspective of supply chain management which focuses on a company's role in a supply chain network.

Remark 3: Even in the same discipline, researchers may have different understandings on the resilience concept. As introduced in the aforementioned part of this section, there are mainly two definitions of resilience in ecology, namely engineering resilience and ecological resilience. The goal of this section is not to discuss and find a unified definition of resilience; yet the goal is to find common features of various definitions of this concept.

Remark 4: In the same discipline, researchers may focus on the resilience of different targets. For example, in supply chain management, Sheffi [2005] studied the concept of resilience from the perspective of one company; while Christopher and Peck [2004] focused on the resilience of a whole supply chain.

2.4 Resilience Engineering: Resilience in Safety Engineering

Resilience and safety may be closely related. Resilience engineering can only be well understood through the understanding of the evolution of the whole safety engineering discipline. Section 2.4.1 summarizes the evolution of safety engineering; in particular, the safety engineering approaches are summarized into three categories with three important properties which reflect humans' understanding of safety: component-based safety engineering, system safety engineering and resilience engineering. Section 2.4.2 and Section 2.4.3 introduce component-based safety engineering and system safety engineering, respectively. Section 2.4.4 reviews the definitions of resilience in the context of safety engineering.

2.4.1 Evolution of Safety Engineering: A Summary

Safety is defined by Mil-Std-882 as "freedom from those conditions that can cause death, injury, occupational illness, or damage to or loss of equipment or property, or damage to the environment" in [Department of Defence of USA, 2000]. Or, in short, safety is "freedom from unacceptable risks" in [Hollnagel et al. 2011]. Risk is defined as "effect of uncertainty on objectives" in [ISO 31000, 2009]. The unacceptable risks may happen to humans, systems or environments, and the harmful event caused by such risks is called accident. An accident is defined as "an unplanned event or event that results in an outcome causing death, injury, damage, harm, and/or loss" in [Ericson 2011].

The evolution of safety engineering goes along with the development of technology and human society and it is driven by the demand from the economy development and also by the methodology progress. Therefore, the safety engineering discipline can be examined from different perspectives, as shown in Figure 2.3. From a technology perspective, safety engineering emerged with the industrial revolution and initially focused on the safety demand of mass production. With the information technology revolution, the social-tech systems, especially service systems, became popular and their safety demands turned out to be important issues of safety engineering. Correspondingly, from a target system perspective, the focus of safety engineering experienced a shift from manufacturing systems to service systems. From an economy perspective, safety engineering witnessed the shift from manufacturing economy to the service economy. From a methodology perspective, safety research had two major schools of methods, quality analysis and quantitative analysis; eventually safety engineering has come into a multi-discipline subject.

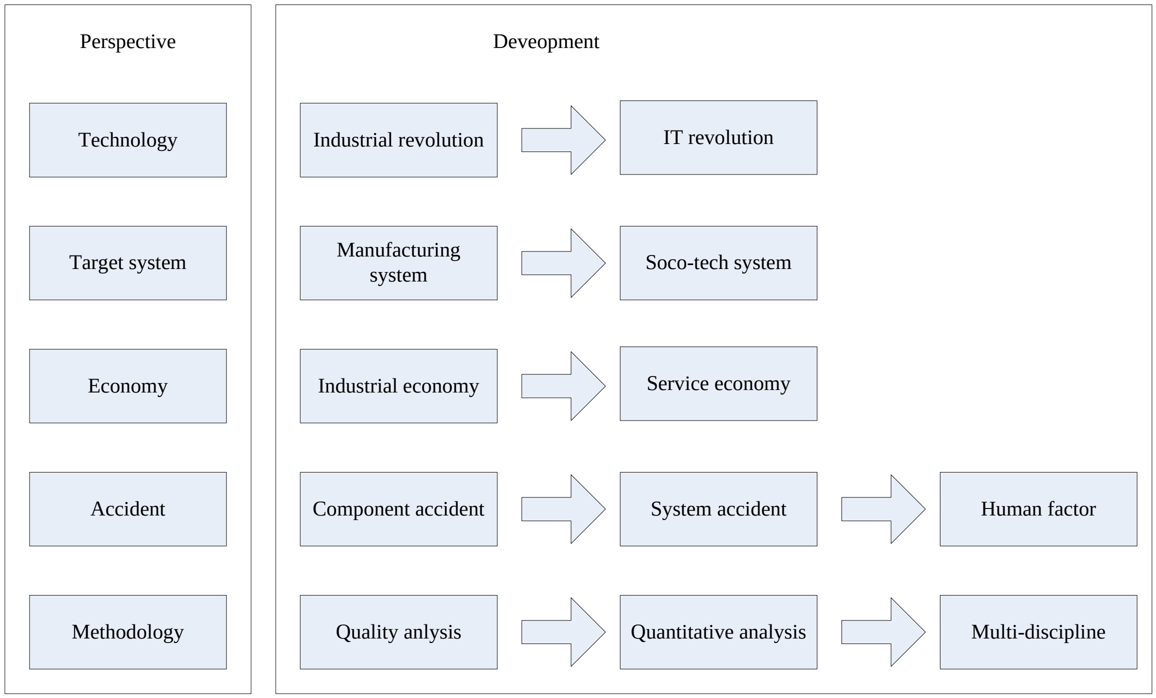

Figure 2.3 Different perspectives on safety engineering evolution

Table 2.7 Evolution of safety engineering

Category	I	II	III
Where accidents come from	Component	System	System
How accidents spread	Linear chain	Complex interaction	Safety-failure balance
Examples models	Sequential accident models, Epidemiological accident models	Sociological models System modeling approaches Cognitive systems engineering models	
Property	Reliability	Robustness	Resilience

Different factors in Figure 2.3 drive the development of safety engineering. As discussed above, safety is defined as "freedom from unacceptable risks" in [Hollnagel et al. 2011]; therefore, the key issue in safety engineering is about understanding risk or accident. Next, the evolution of safety engineering is discussed corresponding to this perspective;

in particular, research on the safety engineering is summarized into three categories: component-based safety engineering, system safety engineering and resilience engineering, as shown in Table 2.7. Details of the three categories are given in this section.

2.4.2 Component-based Safety Engineering

The traditional understanding of safety follows a linear logic and views the safety performance of a system as the sum of the safety performance of each component, which could be called as component-based safety engineering. The earliest concept with this understanding may be "the axiom that a chain is no stronger than its weakest link is one with essential mathematical implications" proposed by Pierce in 1926, which resulted in the focus on the critical component of the system [Verma et al. 2010]. From such a point of view, a system's safety performance could be understood by two classes of accident models and an important system property. The two classes of accident models are sequential accident models and epidemiological accident models, which try to examine how the accidents spread among the components and further lead to the failure of the whole system. The system property is reliability.

2.4.2.1 Sequential accident models

The earliest and classical way to understand the accident is an event chain thinking, which leads to a class of models, called sequential accident models, or event-based accident models [Qureshi 2007, Zio 2009, Verma et al. 2010]. These models consider that an accident happens as the outcome of a chain of discrete events in a temporal order; see [Qureshi 2007, Hollnagel 2002]. One of the earliest models with this thinking is the Domino theory in [Heinrich 1931], which explains accidents as the result of domino effect with five factors, as shown in Figure 2.4 [Ferry 1988, Qureshi et al. 2007]. There are other popular models or approaches that belong to this class, such as Accident Evolution and Barrier (AEB) model [Svenson 1991, 2001], Fault Tree Analysis (FTA), Failure Modes and Effects Analysis (FMEA), and Event Tree Analysis (ETA) [Qureshi et al. 2007, Verma et al. 2010]. Sequential accident models work well on the simple systems; therefore, it is also called simple linear system models [Lundberg et al. 2009].

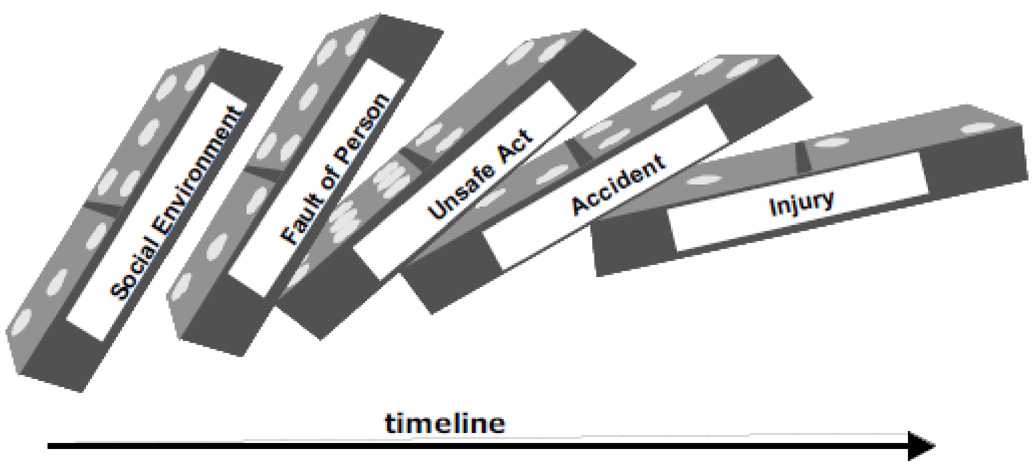

Figure 2.4 Domino model [Adopted from Qureshi et al. 2007]

2.4.2.2 Epidemiological accident models

This class of models considers accidents as similar with the spreading of a disease; in particular, both apparent factors and potential factors that happen to exist together in space and time lead to the accidents [Hollnagel 2002, Qureshi et al. 2007, Lundberg et al.

2009]. The earliest epidemiological approach was proposed by Gordon [1949] to address the accident as the result of the interacting process of three factors, agent (e.g. faulty ladder, cold), host (e.g. food discipline) and environment (e.g. terrain, management of troops) [Lundberg et al. 2009]. The most well-known model of this class is the Swiss chess model proposed by Reason [1990], as shown in Figure 2.5 [Qureshi et al. 2007]. This model considers that a complex system has some defences and barriers and the accident is the result of interactions of weakness, or holes (both active and latent) in these barriers [Reason 1990, Perneger 2005]. The epidemiological accident models work well on complex systems; therefore, it is also called complex linear system models.

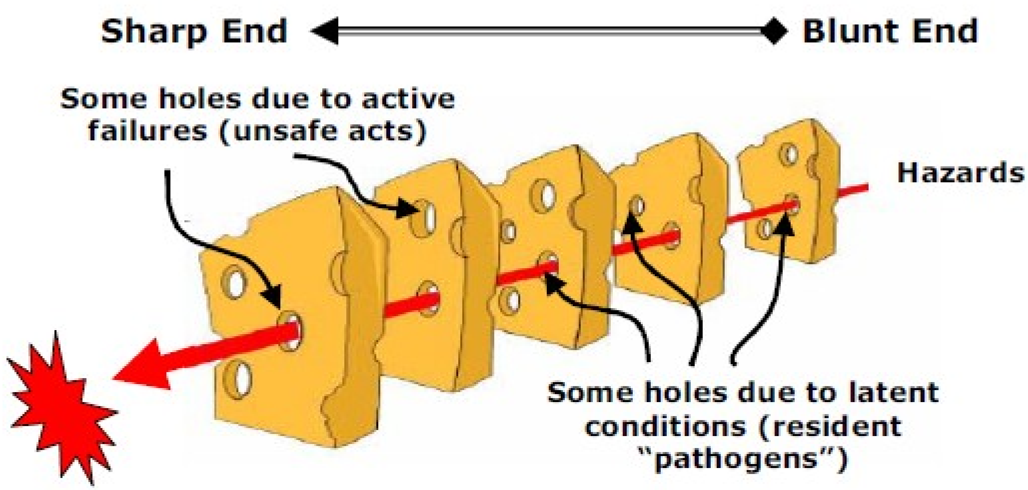

Figure 2.5 Swiss cheess model [Adopted from Qureshi et al. 2007]

2.4.2.3 Reliability and reliability engineering

(1) Reliability Concept

Both of the two classes discussed above follow the linear logic which focused on accidents spreading among the components. As an important safety property, reliability is also introduced into engineering field with the same perspective, which is usually used to examine a system's safety performance.

Reliability is defined as "the ability of a system or component to perform its required functions under stated conditions for a specified period of time" [Zhang 2007, Verma et al. 2010]. Quantitatively, reliability can be represented by the probability of no operational interruption during a given period of time [Birolini 2010]. A reliable system may have redundant parts that can fail and be repaired. The concept of reliability thus applies to both non-repairable and repairable items [Birolini 2010].

It is noted that reliability and safety are different concepts. In some situations, safety and reliability could be viewed as the same properties, especially for simple systems; however, in some special situations, especially for complex systems, they are not identical. In other words, a safe system may be not reliable and a reliable system may not be safe. The main reason that a reliable system is unsafe is that some components or subsystems are able to correctly implement their requirements; however, these requirements are unsafe for the system or incomplete to deal with expected or unexpected conditions [Leveson 2011]. A well-known example is the loss of the Mars Polar Lander, which was attributed to noises [Leveson 2000]. That means: the Mars Polar Lander was reliable, but it was not a safe system.

(2) Reliability Engineering

Reliability, as an important property of the system, has been well discussed in the current literature from an engineering point of view, which forms a discipline called reliability engineering. As pointed out by Leveson [2000], "the reliability engineering approach to safety rests on the assumption that accidents are caused by component failures."

From a system’s structural point of view, reliability engineering has three main aspects: component reliability, system reliability and reliability of repairable component and systems [Modarres et al. 2010]. The component reliability analysis is based on probability and statistics theory and mainly deals with (i) different distributions of component reliability, (ii) component reliability models, and (iii) reliability distribution estimation. The system reliability analysis methods are mainly the reliability block diagram method, fault tree and success tree method, event tree method, master logic diagram and failure mode and effective method [Modarres et al. 2010].

It is noted that the component and system reliability discussed above are largely applicable to non-repairable items. Reliability of repairable component and systems considers different models, such as homogeneous process model, renewal process model, general renewal process model to determine the failure characteristics of items and their reliability. The item here may refer to a system, subsystem, or component. The important assumption for the repairable reliability is that the time to repair is assumed to be negligible [Modarres et al. 2010, Birolini 2010].

From a system's life-cycle point of view, system reliability is examined from different phases, such as reliability at a design phase, test for components and assemblies, maintainability analysis, quality control and reliability test, and reliability of the production phase [Birolini 2010].

Reliability of a service system has also been investigated in the literature. For example, Billinton and Allan [1996] gave a systematic reliability evaluation for power systems; in particular, they further decomposed the reliability of a system into two concepts, adequacy and security. Adequacy refers to system's performance related with static conditions; while system security refers to system's performance related with inner disturbances [Billinton and Allan 1996]. These two concepts, adequacy and security, were examined separately in [Billinton and Allan 1996].

2.4.2.4 Brief conclusion

The understanding of safety by the component-based safety engineering approaches can be summarized into two simple conclusions: (1) A system fails because its components fail; and (2) The accidents spread among the components following the linear principle.

2.4.3 System Safety Engineering

System safety engineering views safety as a problem of the system and uses the system theory and approach to deal with the safety issues. As such, a system's safety can be studied by three schools of accident models and an important system property. The three schools of accident models are sociological accident model, systems model, and cognitive engineering model. The system property is robustness.

2.4.3.1 Sociological accident model

The sociological accident model focuses on the analysis of root causes of accidents of a system. Two typical examples are normal accident theory (NAT) [Perrow 1984, 1999] and high reliability organizations (HRO) [Roberts 1990, LaPort 1996]. Perrow's normal accident theory (NAT) considers that accidents of technological systems are inevitable [Perrow 1999]. He further considered two dimensions of a system's structure (i.e., interactive complexity and loose/tight coupling) as the main cause for a system's accidents; his view of accidents is a top-down view. HRO theory examines system safety from the organizational perspective. Although HRO put emphasis on the organizational factors of system safety, it still has a bottom-up view of accidents. It is noted that both NAT and HRO seem to say that redundancy is the only way to handle accidents [Marais et al. 2004].

2.4.3.2 System model

From 1990s to 2000s, a thought that safety should be viewed as a property of the system emerged among many others, some important studies refer to [Rasmussen 1997, Hollnagel 2002, Woods and Cook 2002, Leveson 2004a]. The emergence of such a systems approach may be attributed to the fact that systems have been more software-intensive and further connected with the Internet making the systems more coupled with uncertain dynamics and more open to new attacks [Leveson 2004b].

Leveson [2000] summarized the primary differences between a systems approach and a traditional safety engineering approach as: "(1) top-down systems thinking rather than a bottom-up fault tracing, reliability engineering focus and (2) focus on the integrated socio-technical system as a whole and the relationships among the technical, organizational, and social aspect." There are two representative models of this system approach school: the hierarchical socio-technical framework proposed by Rasmussen [1997] and STAMP (System-Theoretic Accident Model and Processes) proposed by Leveson [2004a].

Rasmussen's hierarchical socio-technical framework views the cause of accidents as a complex process at different levels, as shown in Figure 2.6 [Qureshi et al. 2007]; in particular, he claimed that a system is more than the sum of its elements and a model of the system for safety analysis can never be built up by a bottom-up aggregation of models, but a top-down system-oriented approach based on control theoretic concept [Rasmussen 1997]. The idea of different levels of subsystems for a social-technological system is not new; but it is Rasmussen that first introduced this idea into safety engineering. His contribution also rests with (1) introducing a control perspective to safety engineering, (2) viewing safety engineering as a multi-discipline study, and (3) safety analysis with a focus on functional abstraction rather than structural decomposition. However, in his framework, there is still the event chain thinking especially at the hazardous process level according to [Leveson 2004a]. It is in the writer's opinion that Rasmussen's approach is difficult to be practiced or implemented due to the ambiguity in his approach.

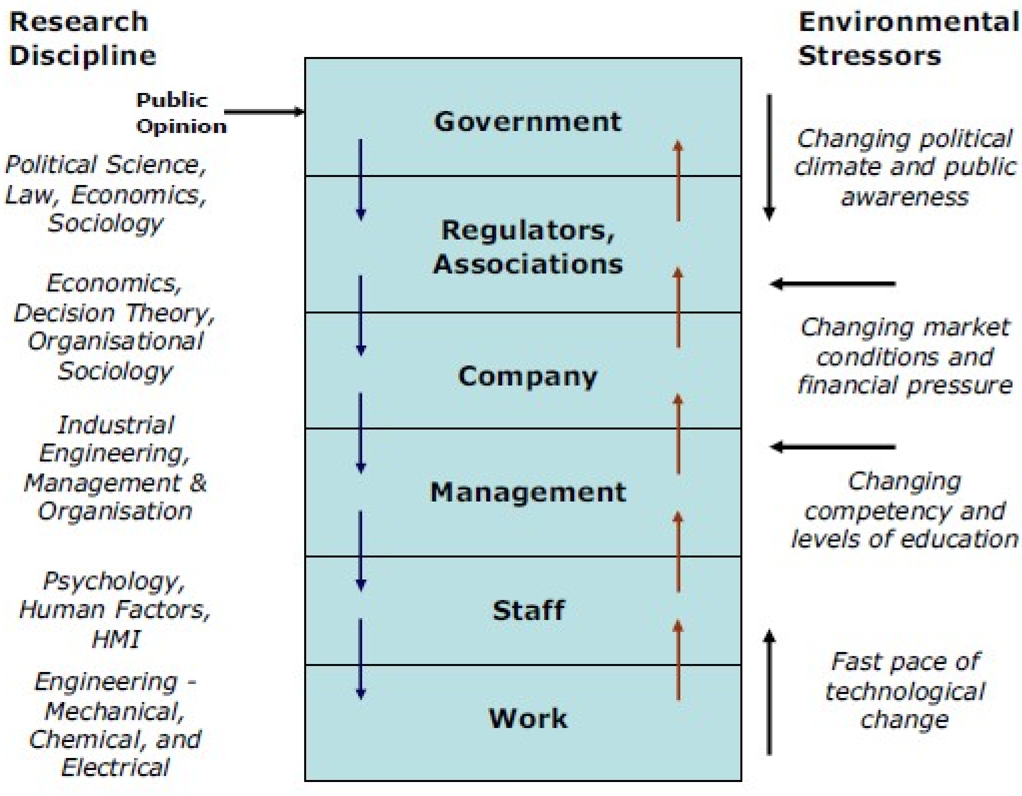

Figure 2.6 Hierarchical socio-technical framework [Adopted from Qureshi et al. 2007]

Leveson's STAMP model was proposed based on the work of Rasmussen, especially the control-theoretic operations [Leveson 2004a]. It was the first time that the safety was stated as the emergent property of a system, though Rasmussen did imply that. It is noted that the emergent property means that it can never be summed up from the performances of individual components. Leveson stated: "system theory is a useful way to analyze accidents, particularly system accidents" [Leveson 2004a]. Thus, the STAMP model takes the safety problem as a control problem; furthermore, the accident is the result of an inadequate control. In the opinion of the author of this thesis, the STAMP model has not made clear about (1) whether the control in the model is a real-time process or just an off-line process, and (2) what the controller is and its action is. As it is known, the contemporary control system theory does not explicitly include any change in the plant as a control action. That said, the controller's output does not include any action or process leading to a change of the plant. This consequently leaves the plant system design and reliability engineering out from the safety concept.

2.4.3.3 Cognitive systems engineering models

The cognitive systems engineering concept was proposed by Hollnagel and Woods [1983] with the trend that computer was more frequently used in the human-machine systems and the systems became more intelligent. A cognitive system views humans as a part of the system and the system is adaptive and able to plan and modify its actions based on the knowledge and models of the system itself and its environment [Hollnagel and Woods 1983]. Cognitive system engineering considers two aspects in design of a cognitive system; one is the man-machine system (MMS) itself and the other is the cognitive task, as shown in Figure 2.7.

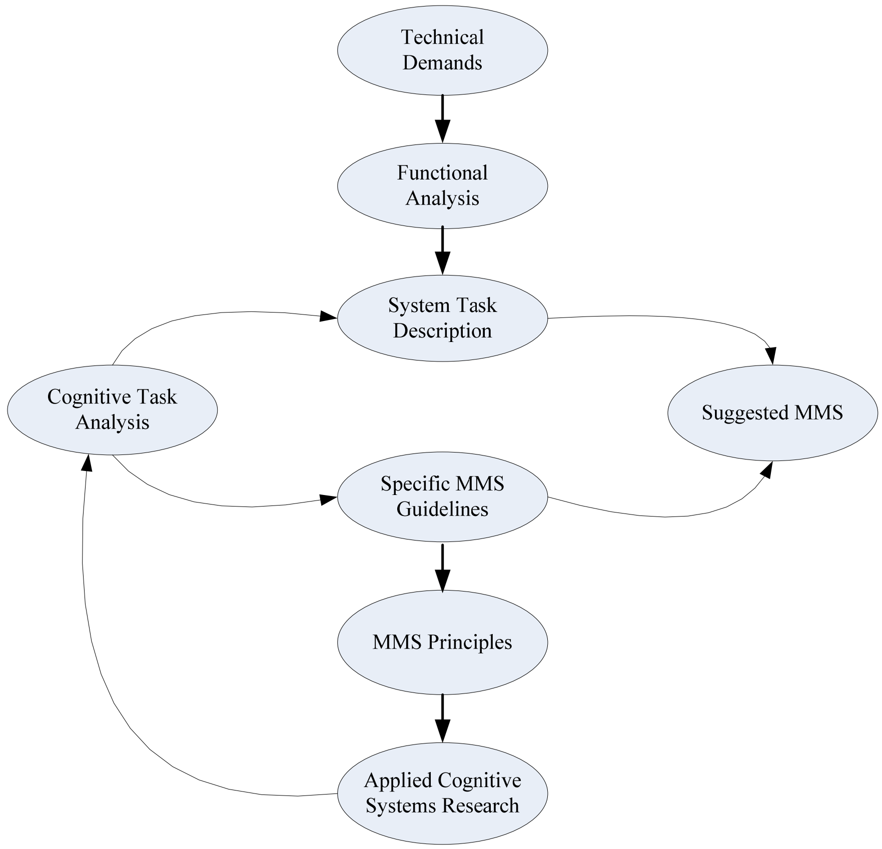

Figure 2.7 Cognitive systems engineering in the design process

The principles of cognitive systems engineering were applied to the safety engineering and two accident models were proposed: the Cognitive Reliability and Error Analysis Method (CREAM) [Hollnagel 1998] and the Functional Resonance Accident Model (FRAM) [Hollnagel 2004]. CREAM is a bi-directional method with both retrospective analysis and prospective analysis of accidents; in particular the analysis process is recursive instead of hierarchical or sequential [Hollnagel 1998, Serwy and Rantanen 2007]. FRAM uses the principle of stochastic resonance to explain complex accidents and to identify risks (i.e., potential failures) in a dynamic system [Hollnagel and Goteman 2004]. In the author's opinion, the cognitive engineering approach is applicable to human-machine system only. The key issue is task allocation between the machine and human in light of safety. The approach does not cover the safety problem of a machine system.

2.4.3.4 Robustness and robustness engineering

(1) Robustness Concept

Robustness is a property of a system related to noises. In particular, robustness is defined as "an ability that allows a system to maintain its functions against internal and external perturbations or noises" in [Kitano 2004, Zhang 2007]. Robustness of a system focuses on how a system is insensitive to noises. In Perrow's work, as mentioned above, interactive complexity is one important cause for a system's accidents. He defined interactive complexity as "the presence of unfamiliar or unplanned and unexpected sequences of events in a system that either not visible or not immediately comprehensible" in [Perrow 1984]. His theory seems to emphasize the noises within the system but to neglect the noises coming from the system's environment that affect the system's performance per se. Design and control are important engineering activities to affect a system's robustness. As such, there are the notions of robust design and robust control in the literature but the two are less integrated. Robustness is a property that is built upon reliability. A system is said to be robust by satisfying the two conditions: (1) the system is reliable in individual components and their connections, and (2) the system is insensitive to noises. Details of the justification of these assertions will be provided in later chapters of the thesis.

(2) Robustness Engineering

The earliest work on the system's robustness is robust design which was developed by Fisher and Yates in 1920s; particularly they used the statistical design of experiments (DOE) method to agricultural system [Fisher 1951]. From the 1950s to 1960s, Taguchi established the foundations of robust design for quality control in process plants [Taguchi 1986]. Taguchi viewed robustness as "a state where a technology, products, or process's performance is minimally sensitive to factors (including aging) causing variability at a low unit manufacturing cost" in [Taguchi 1986]. He further explained that there are two types of quality in a product or process: customer quality and engineered quality. Robustness engineering is associated with engineered quality. He divided a product design process into three stages: system design, parameter design and tolerance design; furthermore, he proposed the signal-to-noise ratio (SNR) as the measure of robustness. The robust design method has been proved to be a successful method in different industrial fields according to [Phadke 1989, Taguchi 1986].

From an engineering perspective, a system's robustness is also examined by other approaches, for example, robust control [Feng and Brandt 1998], robust planning [Tizghadam and Leon-Garcia 2009] and robust management [Wolkerstorfer et al. 2010, Marco et al. 2010]. In the current literature, the term "robustness engineering" is only used in the literature of robust design; particularly, they called robust design as robust engineering or robustness engineering. However, design, control, planning and management, are all engineering activities, and they should be included in the scope of robustness engineering. Therefore, in this thesis, robust engineering has a more general sense than that in the literature with consideration of all the robust engineering activities mentioned before.

2.4.3.5 Brief conclusion

Safety is an emergent property of the whole system, and accidents happen due to complex non-linear interactions among components and external and internal disturbances.

2.4.4 Resilience Engineering

2.4.4.1 Understanding safety through the resilience concept

The concept of resilience in the context of safety engineering was proposed by Hollnagel et al. [2006]; in particular, resilience engineering uses the resilience concept to describe the safety performance of a system. Resilience is also viewed as an emergent property of a system.

Resilience engineering could be viewed as an extension of system safety engineering, as Hollnagel's resilience engineering also views safety as a system's issue, and makes further step on the understanding of safety. The basic concepts and precepts of resilience engineering have been addressed by different researchers [Hollnagel et al. 2006, Patternson et al. 2007, Hawks and Reed 2006, Bursztein and Goubault-Larrecq 2007, Zhang 2007]. Different from component-based safety engineering and system safety engineering, resilience engineering does not focus on how accidents happen and how accidents may lead to the failure of a system; there is not any particular accident mechanism in the notion of resilience for the system [Hollnagel et al. 2010]. There are many factors that affect a system's safety, and there is no clear difference between the factors leading to safety and factors leading to failure. Resilience engineering defines safety as "an ability to succeed under varying conditions" in [Hollnagel et al. 2010]. From a resilience engineering perspective, safety and failure are not two absolutely different states; in particular, the safety of a system refers to "a state of the balance between the safety factors and failure factors which are the further outcome of the core process of the system" in [Hollnagel et al. 2010].

2.4.4.2 Definitions of resilience in the context of safety

The definitions of resilience in other disciplines have been presented and discussed in Section 2.3. This section focuses on the definitions of resilience in the context of safety; in particular, the definitions of resilience are classified into three categories, as shown in Table 2.8.

The first category views the resilience of a system as the similar thing with reliability or robustness. For example, Victoria Transport Policy Institute [2010] used the concept of reliability to define resilience. Another example is the work of Bongard et al. [2006], where the resilience of a system was considered as the ability of a system that can recover from an unexpected damage and a continuous self-modeling method was proposed to enable a machine system to autonomously recover its own topology from unexpected change by reconfiguring the system Note that their system does not have any built-in redundancy. The resilience concept in Bongard et al. [2006] is similar with the second category of definition of resilience; however, they also used robustness to label such a property without any differentiation from resilience. There were also some work that used the concept of resilience, e.g. [Rosenkrantz et al. 2005, Wang and Ip 2009]; however, their concept of resilience seems to make more sense of reliability. This category of definitions of resilience has not shown a clear understanding of resilience, which leads to the confusion of the three properties, reliability, robustness and resilience. It is necessary to clarify their relationships.

Table 2.8 Definitions of the resilience in the context of safety

Category	Definition	Literature
I	Reliability or robustness.	Victoria Transport Policy Institute 2010, Bongard et al. 2006
II	The ability of a system to recover to meet the demand from a partial damage.	Zhang 2007
III	The intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances so that the system can sustain required operations under both expected and unexpected conditions.	Hollnagel et al. 2010

The second category of definition of resilience followed the resilience concept discussed in Section 2.3. The typical one was proposed by Zhang, as shown in Table 2.8. Zhang and Lin presented a comparison of different definitions of resilience and showed a situation where there was no agreed definition for resilience [Zhang and Lin 2010]. Their conclusion was similar with the result of Section 2.3, as the resilience of a particular system depends on its particular features. However, this definition does not work for the service systems. For example, the initial reason of the India power system failure, as mentioned in Chapter 1, was the large increased demand which cannot be met by the system. In such a case, the system's failure was not caused by any damage.

The third category of definition of resilience was proposed by Hollnagel et al. [2010], as shown in Table 2.8. Hollnagel further proposed four cornerstones of resilience as: "knowing what to do, knowing what to look for, knowing what to expect, and knowing what has happened" [Hollnagel et al. 2010]. This definition actually covers the traditional concepts of safety, reliability and robustness, and makes further extension; in particular, resilience concept deals with the damaged situation – a point of resilience stressed by Zhang [2007]. Therefore, this category of definition of resilience is an all-inclusive definition, which is similar with the third category of definition of service system discussed in Section 2.2. The all-inclusive feature of the definition further makes different properties of a system, such as reliability, robustness and resilience, vague. Thus, it also does not work for a service system.

2.4.4.3 Related work

Resilience engineering mainly concerns socio-technical systems [Allenby and Fink 2005]. In a socio-tech system, the human plays an important role. On one hand, the human is a part of the system, or the human could be viewed as a component of the system; the human may make errors or may not be available in time. On the other hand, the human is an important resource for the recovery of the possible damage of the system [Wang et al. 2013b]. The research in the current literature can be summarized into two categories, (1) how to measure the resilience of a system, and (2) how to enhance the resilience of a system.

Regarding the measurement of resilience, Rosenkrantz et al. [2005] used fault tolerance to measure the resilience, which considered the number of node failures and edge failures while the network can still keep its normal function. Wang and Ip [2009] proposed an approach to evaluate the logistics network resilience based on the redundant resource and reliability of nodes and edges. Both of these studies mentioned above focused on what is “left” in the system after damage – or called “remaining” part of the system; in particular, their example systems keep normal functions without interruptions. They measured the resilience of a system through the physical redundancy or functional redundancy, which is the case of reliability. Lhomme et al. [2011] described a shared diagnosis approach between different urban service network managers based on the scientific assessment of failures and interdependencies. However, they only considered the technical network in the assessment of failures without any consideration of service itself. Kanno et al. [2011] proposed a method to assess the resilience of a service system during a disaster. They considered both service activities and infrastructure into the method; in particular, their services are only related with human activities. However, the way that service is provided to meet demands is related not only with human’s activities but also other substances on the infrastructure [Wang et al. 2013b]. Ferreira et al (2010) introduced an approach to measuring the resilience by means of a questionnaire. Allenby and Fink [2005] considered that the evaluation of resiliency only makes sense in terms of an identified system and particular challenges.

Regarding the enhancement of resilience for a system, one of the important contributions on resilience was made by Leveson et al. [2006] who used their previous approach, STAMP, to enhance resilience with the following features. First, they viewed the whole problem as a control problem. As such, there is a goal, and resilience is associated with drifts in the behaviour of the system away from that goal. Second, they employed the system dynamics approach [Sterman 2000] to model the dynamics of the system. Third, they developed safety resource allocation (SRA) system. SRA makes decisions on resource allocation to diminish identified drifts in the system. Their approach has not explicitly represented the domain structure and ontology of the system; i.e., they have missed a so-called domain model. It should be noted that the domain model corresponds to what we called “conceptual object model”, which is very important to make an intelligent system. Another limitation of their approach is that the problem of vulnerability prediction has to be modeled as a time-driven continuous problem, which is due to the fact that the system dynamic approach they applied eventually results in a continuous time driven equation. As far as the control of the system toward recovery of the remaining system to its original function, they did not explicitly describe how this can be done.

From a system’s life-cycle point of view, the system’s resilience is examined from different phases, such as design, planning and control (management). In the current literature, the resilience is mainly examined from a particular phase, for example, resilient design [Ulieru 2007] and resilient management [Hall-May and Surridge 2010, Jung and Pedram 2008]. This is also the case for the previous concepts, service system, service-oriented system, reliability and robustness.

Another deficiency of the current research on measurement and enhancement or improvement of resilience is that the confusion on the understanding of resilience may lead the work to a wrong direction, for example, using the reliability measurement for resilience [Najjar and Gaudiot 1990, Wang and Ip 2009].

2.4.4.4 Brief conclusion

The understanding of safety through the resilience engineering approaches can be summarized into two conclusions: (1) safety is the balance between the outcome of safe factors and the outcome of failure factors, (2) resilience engineering focuses on how to measure and improve the system's resilience; however, the current work in the literature is far insufficient due to the unclear understanding of the system's resilience.

2.5 Conclusions

This chapter reviews two topics: service system and resilience engineering.

(1) Regarding the service system, a similar concept, service-oriented system is also introduced and discussed. However, it appears that an identity of the service system is still not clear; in particular, the key features of service system with regard to safety performance are not clear.

(2) Regarding the resilience concept, its development and definitions in different disciplines are discussed for further understanding the concept in the context of safety engineering.

(3) Regarding the resilience engineering, it is discussed based on the introduction of the whole evolution of the safety engineering discipline. Three categories of safety engineering approaches, namely component-based safety engineering, system safety engineering and resilience engineering are discussed. Three important safety properties related with these three categories, namely reliability, robustness and resilience are also discussed separately. However, the summary of the resilience definitions indicates that the relationships among the three R properties of systems especially service systems still remain unclear; furthermore, the current resilience understandings do not work for the service system. The awareness of these issues is very important, as they are the keys which lead to the proper efforts on the measurement and improvement of the resilience.

Thus, the next two chapters contribute to the better understandings of service system and its property of resilience.

CHAPTER 3
TOWARDS A BETTER UNDERSTANDING OF SERVICE SYSTEMS

3.1 Introduction

In Section 2.2, several important definitions of service system in the current literature were introduced, and the difficulties with these definitions in terms of distinguishing service system from other systems were discussed as well. The two main problems in the current definitions of service system are: (i) they tend to focus on the concept of service and miss a system's perspective of service, i.e., service system, and (ii) they tend to be all-inclusive. Indeed, there is still an on-going debate on what service system is and why there is a need to distinguish a service system from other systems; note that the two questions in debate are inter-related. A well-known example which may reflect the debate is that Fortune Magazine has given up its attempt to differentiate service firms from manufacturing firms since 1993 after many years of publishing both the Fortune Industrial 500 and Fortune Service 500 [33]. The Fortune's movement may lead to the view that there is no need to consider service as an independent economy entity.

This thesis argues that service system has its own identity and subsequently, provides a note on the two phenomena as mentioned elsewhere, namely (1) change in world job distribution over different sectors of economy in the 2007 report by the International Labor Organization, and (2) no differentiation of service firms from manufacturing firms in Fortune Magazine since 1993. It is noted that the existing definitions of the service system in literature were listed in Section 2.2.

This chapter is thus organized in the following. Section 3.2 describes three systems, namely manufacturing, agricultural and product systems to prepare for the comparison of them with service system in order to study the identity of service system. Section 3.3 presents a unified definition of service system. Section 3.4 gives a discussion of the difference of a service system from a manufacturing, agricultural and product system. Section 3.5 concludes the chapter with an analysis of the underlying reason behind the aforementioned two phenomena.

3.2 Agricultural System, Manufacturing System and Product System

An agricultural system produces plants and food. The generic architecture of an agricultural system is such that it includes humans (farmers), equipment, soils and organization, where the organization performs the administration and management functions. Examples of the agricultural system are the rice farm system, pig farm system and so on.

A manufacturing system makes engineered products. The generic architecture of a manufacturing system is such that it includes humans (workers), equipment, and organization. It can be seen from this definition that a manufacturing system does not have elements provided by nature, a point which allows the manufacturing system to be distinct from the agriculture system. Examples of the manufacturing system are the iron and steel industry system, chemical plant system and so on.

A product system is a tangible entity that performs the conversion and/or transfer of motion, force and power. The generic architecture of a product system is such that it does not include humans but humans are certainly a user of a product system. Examples of the product system are car, cell phone and so on.

The next section will give a new definition of service system. The three systems discussed above will serve a test-bed to justify whether the new definition can differentiate a service system from a manufacturing system, a product system and an agricultural system.

3.3 A Unified Definition

The writer defines service and service system as follows. A **service** is a function that is achieved by interactions between a human and an entity under a protocol. A **service system** or organization or firm consists of three subsystems: (i) an infrastructure, (ii) a substance, and (iii) a management to directly meet the demands of humans (who are also defined as **consumers)**. The infrastructure is of a network, and substance "flows" over an infrastructure. The management plays roles such as coordinating, leading, planning and controlling, which are applied to both the infrastructure and substance systems.

Remark 1: The key feature of service system is that the system can meet the customers' demands directly, not through other systems. The sense of a service lies in that a human's status or state is changed to meet his or her need directly by operation of a system. Production of goods needed by a human is not a service, as it focuses on the event from a “raw” material to a “product” (i.e., goods.); the demand of humans or the customers is met by goods, not directly by a production process of goods. Therefore, an agricultural or a manufacturing system is not a service system. A transportation system which moves goods from one place to another where the customer receives the goods is a service system, as it focuses on the change of goods in its location and time to meet a human’s need.

Remark 2: The demands of the customers and services offered by a service system are interactive. The customer in this definition refers to the human not any other entity say company. The demands of humans change and the service systems need to meet the demands. This implies that the demands of humans have impact on the output of the service systems or the demands could be viewed as a part of the input of the service systems. Note that the services supplied by the service systems may also affect the demands of customers. For example, some special rules may be applied to a transportation system to constrain the use of some roads, which will then change the demands of customers. Another example is the power supply system which may change the price of electricity and thus affect the demands of customers.

Remark 3: The new definition covers both the structural and functional aspects of a service system as well as the aspect of operation management. This definition does not highlight the term ‘organization’, which differs from the others’ in the literature, e.g. the one of [Pinhanez 2009]. The new definition is considered that the concept of organization has been implied in the concept of the system. Organization refers to putting things together, which is exactly the nature of any system.

Remark 4: The service system as defined is structurally generic and functionally general. The phrase ‘structurally generic’ implies that the concepts and features captured of service system are common for all types of service systems. The phrase ‘functionally general’ means that all kinds of services such as transportation of goods, health service and travel advisory agency are covered.

Remark 5: The substance can have four generic types: material, human or animal, energy, data or signal. Data further make sense for information or knowledge depending on a service’s receiver especially on the effect of the service to the receiver per se. According to Zhang [1994], when a data is used to inform a user of something that is otherwise not available to that user, the data is called information; when a data has an effect on a user such that the user enables to do something, where ‘to do something’ is otherwise not possible, the data is called knowledge. The infrastructure includes both equipment and humans.

Remark 6: A resource is a physical or cognitive entity with limited availability and accessibility that needs to be consumed to obtain a benefit from it. In the context of the service system as defined, a resource could refer to both the infrastructure and substance; that is to say, there may be an infrastructure resource or a substance resource.

Remark 7: That the structural aspect of a service system puts emphasis on network is to capture a situation where the points where resources “meet” substances typically have many and change with respect to time and space. For instance, in emergency evacuation, transportation tools are resources to be put into an infrastructure system (road, bridge, etc.) [Wang et al. 2013b]. Where tools are put on infrastructure is determined depending on situations, and this decision is in fact a part of the management decision. It is required that a service system must be a network structure and have many “ports” linking to customers and many “ports” to which resources can be connected. With the proposed definition, service system and networked service system are on the same page.

Remark 8: A protocol is an agreement or constraint between service providers and service demanders. A chair in a meeting room is not a service system, as there is no protocol specified per se. A chair is a product. However, a meeting service department is a service system, as there must be a protocol available between anyone who wants to use a meeting room. In this context, a chair in the meeting room becomes a part of the meeting service system, through which a part of the services, e.g., provision of the seating function, is achieved.

As discussed in Section 2.2, the most popular definition of service system, namely the third category of definition which was proposed by Spohrer et al. [2007, 2008], focuses on the value-coproduction configuration. However, the essence of economy is value production or co-production which meets the people's demand. Along this line of thinking all economies could be called service economy [Vargo et al. 2008], which does not sound reasonable. The proposed definition in this section clearly resolves this issue through narrowing the scope of the concept. In the definition here some economic sectors or systems meet customers' demands directly; while others meet customers' demands indirectly, in particular via a manufacturing or agricultural system. The former systems are service systems and the latter systems are either manufacturing systems or agricultural systems.

Several examples of the service system based on the new definition can be illustrated. A telecommunication system is a kind of service system, where infrastructure is the equipment such as cable, switch and so on [He 2008] and substance is the data. Management is the data flow control policy. A travel agent system is a kind of service system, where infrastructure in this case is human agent and transportation equipment, and substance is tourists. Management is planning and scheduling of the equipment and tourist. Hospitals and medical centers are a kind of service system, in which the infrastructure consists of both medical professionals and equipment, and substance consists of patients, and management makes sense for both the infrastructure (medical doctors and equipment) and substance (patients) systems.

3.4 Relationships to Other System Concepts

In the following, the difference of a service system (based on the proposed definition) from the three other systems (i.e., agricultural system, manufacturing system, and product system) is further explained.

A product system is not a service system because a product system is not of a network, does not have many ports to many customers, and further a product system has only an infrastructure system only. For instance, a cellular phone alone is not a service system, as it has no substance; however, when it is integrated with a telecommunication system, it carries substance, which makes it as a part of the telecommunication system that is a kind of service system. A single and integral subject or system is not a service system even if this subject or system provides a service to a customer. This is because such a system is not of network when a service is provided. For example, a human who provides a massage service is not a service system.

More recently, the concept of product-service system (PSS) has emerged to offer a seamless integration of products and services. The use of and ownership of a PSS are separated; in particular, a PSS is developed by the sale of its use instead of its ownership [Mont 2002]. A PSS can be viewed as a specialized product system owing to its product effect, and a PSS can be viewed as a specialized service system owing to its service effect. Therefore, a PSS is neither a product system nor a service system but a system composed of a service system and a product system.

A manufacturing system or an agricultural system is not a service system because they focus on transforming “raw” materials to “products”, through which they meet customers' demands. There is a view that a manufacturing system or an agricultural system may include a transportation system from a point of view of customers in [Li 2004]. While such a view may be meaningful to the end customer, the manufacturing system and transportation system are different in their ways to meet customers' demands; furthermore, they are subject to different principles in design and operation management. This means that the two systems need a separate attention at a point where design and management are concerned. It is more meaningful to have an integrated system such as "manufacturing-service", "product-service" systems, and "service-manufacturing-product". These integrations have not eliminated the identity of each member system but rather bring together the member systems for a better business performance. The rationale behind these integrated systems is that features of member systems may have some coupled effects on the customer satisfaction and customer-perceived value. These integrated systems may be viewed as a kind of service system or a kind of product or manufacturing system in some contexts.

3.5 Conclusion with Further Discussion

This chapter presented a unified definition of service system. This was achieved by taking a view of service system from both the structural and functional aspects. The goal of developing a unified definition of service system was that the definition should be generic, general, and functional in addition to provision of an identity of the service system, distinguishable from the agricultural system, manufacturing system, and product system. It appears that the proposed definition has achieved this goal. Furthermore, with this definition, the two phenomena mentioned before can be explained.

For the phenomenon that Fortune Magazine has not differentiated the manufacturing firm from the service firm since 1993, the underlying reason is that the generic architecture of the service system and that of the manufacturing system are of no difference, yet the difference being at the instance level (e.g., number of humans, degree of automation, type of machines, etc.). It is noted that a unified definition of the manufacturing system, similar to the unified definition of the service system can be found in the work of [Zhang and van Luttervelt 2011]. Consequently, a company may have both the manufacturing business and the service business, as both can produce values. Assessment of a company from its value can hardly then distinguish the value created by its manufacturing system or its service system. As such, with the attention to economic nature only, there is no need to separate service from manufacturing system.

For the phenomenon of a dramatic increase of jobs percentage in service sectors, first this may come at some expense of the decrease of jobs in manufacturing systems and second this may be caused by inclusion of the service system in many firms which were originally 100% manufacturing firms in the last decades. This second point is further related to the need of coping with the decline of employment in the manufacture sector, especially in some developing nations (e.g., China) in the last decades owing to the side effect of enterprise reform. Further, dissolving one large service system which previously worked for several manufacturing firms into several small service systems incorporated by these manufacturing systems will introduce redundancy in small service systems, thereby increasing the total number of jobs in service sectors. The third reason for the increase of jobs percentage in service sectors may be related to the missing of a unified definition of the service system, a proposition hold by this thesis. Indeed, the missing of a correct definition of service system can lead to a situation where all types of systems are viewed as service systems. Consequently, manufacturing systems are thus viewed as service systems, and naturally the number of jobs in the service sector will increase, while the number of jobs in the manufacture sector will decrease.

CHAPTER 4
DOMAIN MODELING FOR SERVICE SYSTEM AND ITS APPLICATION

4.1 Introduction

Modeling of ontology or domain of service system is an essential step to more effectively manage service systems across different firms. There are two major roles for a domain model: a common language for different humans to communicate with each other about a domain (service in this case) and a tool to allow the computer system to understand the semantics of a domain (service in this case). As such, a domain model will facilitate both distributed human-centered decision making and computer-assisted decision making in design and operation of a system in the domain. A unified approach to domain modeling is still absent in the service and service system. A justification of this observation will be provided in Section 4.2.

The objective of this chapter is to propose a framework for domain modeling for service system. It is noted that the focus in this chapter is on the framework for modeling or generic model [Zhang et al. 1993, Zhang 1994], instead of a concrete model of service systems. That said; the work presented in this section will only provide a set of modeling building blocks tailored to service system. It is expected that with this framework, a generic service system domain model can be established, followed by specialized service system domain models such as enterprise information systems or transportation systems. To achieve this objective is to apply a general domain modeling tool called FCBPSS, which was developed by Lin and Zhang [2004, 2005] upon a careful analysis of various similar modeling tools which are only based on three concepts: function, structure, and behaviour.

This chapter is further organized as follows. In Section 4.2, a literature review on domain modeling of service system is presented. In Section 4.3, the FCBPSS framework is briefly described. In Section 4.4, the framework for domain modeling for the service system using the FCBPSS is proposed. Section 4.5 discusses the effectiveness and usefulness of the model. Section 4.6 applies the proposed model to examine the generic variables of service system. Section 4.7 concludes this chapter.

4.2 Related Work

Modeling of service system is known as an important topic due to ever increased complexity of the modern service system. A systematic modeling approach to service systems requires domain modeling first. Domain modeling is considered to be similar to ontology modeling in this thesis, and they further depend on the definition of domain of an application – service system in this case. There are only a few studies in literature on domain modeling in the context of service system. Besides, they are all based on the problematic definitions of service system; see discussions in Section 2.2 and Chapter 3, respectively.

Stanicek and Winkler proposed a conceptual model for the service system [Stanicek and Winkler 2010]. It is noted that a domain model is a kind of conceptual model. The limitations of their model are: (1) the model is based on an extension of the service system definition proposed by Spohrer et al. [2007], which has difficulty in providing an identity of the service system, and (2) the model is much focused on the service delivery not on the constituent elements of the service system. The proposed framework in the present chapter will be shown to overcome these shortcomings. Further, the proposed framework also overcomes some shortcoming with the model of Stanicek and Winkler [2010] in that the proposed framework represents the whole generalization/specialization lattice of conceptual modeling of the service system, whereas Stanicek and Winkler's model is at a certain level of this lattice.

Dinh and Thi [2010] presented a conceptual framework for service modeling in a network of service system and used simplified UML (Unified Modeling Language) for the meta-model in the framework. Their work was also based on the definition of service system proposed by Spohrer et al. [2007], which, however, involves some conflict. As discussed in Section 2.2 and Chapter 3, the definition of service system by Spohrer et al. has a very general scope and views that individuals, families, firms, nations, and economies are all instances of the service system [Spohrer et al. 2007]; however, the framework proposed by Dinh and Thi [2010] considered service system into a narrow scope which views network as a higher level of service system. Further, UML is a tool which is based on object-oriented (OO) paradigm. The OO modeling approach is restricted in its expression power to real world phenomena and activities, as it flats them into a framework which has only two levels: object and its method or end and means.

In general, as far as the domain modeling tools are concerned, the current tools employed for domain modeling in the context of service systems are at most based on the OO paradigm. These tools are not natural in modeling of rich real world semantics in service systems. Specifically, they are poor at capturing the domain semantics in why, how, where and when a means can achieve an end; the modeling approach based on the method in OO is just too general to capture these semantics. The FCBPSS [Lin and Zhang 2004, Zhang et al. 2005] to be introduced in the next section can capture these semantics.

4.3 The FCBPSS Framework for Domain Modeling

The FCBPSS framework is a methodology as well as tool to develop a conceptual or domain model of the dynamic system proposed by Lin and Zhang based on the FBS framework [Lin and Zhang 2004, Zhang et al. 2005].

The FBS framework was initially proposed to increase the intelligence of computer systems for fault analysis [De Kleer 1984, Kuipers 1984]; in particular, the FBS in their work refers to the Function-Behaviour-State (FBS) model. The FBS tool has been successfully applied to different areas such as engineering design, simulation and diagnosis and software engineering in [Ulrich and Seering 1988, Umeda et al. 1990, Umeda and Tomiyama 1995, Umeda et al. 2005, Kruchen 2005].

The FCBPSS framework was proposed by modification and extension of the Function-Behaviour-State (FBS) framework to have more layers of concepts [Lin and Zhang 2004, Zhang et al. 2005]. The FCBPSS framework has a set of key concepts: (1) structure, (2) state, (3) behaviour, (4) principle, (5) function, (6) context, (7) relationship among concepts (1)–(6), and (8) system decomposition. Figure 4.1 shows these concepts and their relations [He 2008]. The definitions of these concepts are referred to the reference [Lin and Zhang 2004, Zhang et al. 2005], while Appendix A of the thesis provides simplified definitions for the convenience of readers.

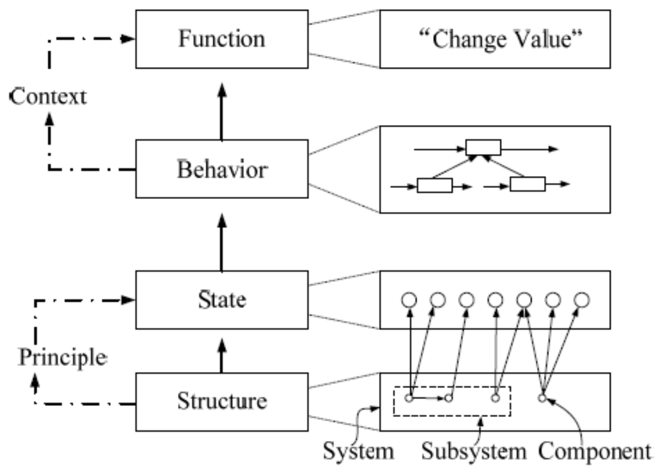

Figure 4.1 Architecture of FCBPSS framework [He 2008]

The next section will present a domain modeling framework for service systems, which is illustrated by several typical service systems, such as enterprise information system and transportation system.

4.4 FCBPSS Model of Service System

4.4.1 System Decomposition

As stated in the proposed definition in Section 3.2, a service system has three subsystems: infrastructure system (IS), substance system (SS) and management system (MS). IS and SS are related to each other by the inherent constraint that the SS depends on the IS or the SS must "flow" within the IS and both IS and SS are under the management of MS. The flow of SS follows certain constraints which could be called "substance flow rules". These rules are derived from a particular service system under investigation. The dynamics of service system is determined by the flow of SS under the constraints of these rules. MS is a body of decisions in its nature; in particular, the MS is designed and implemented based on objects to be managed: IS and SS. The FCBPSS model of service system will focus on the IS and SS. In the following, these models are presented.

4.4.2 Structures of the IS and SS

(1) Structure of the IS

The structure of the IS refers to a network of components including both physical entity and human. This network can be further represented by graph formalism to facilitate the modeling. In the case of the public transportation system, node is defined for place and arc is defined for lane. Suppose a particular transportation system has M nodes and N directed arcs. A transportation system can be represented as a directed graph denoted by

G and $G=(N,A)$, as shown in Figure 4.2. In the case of an enterprise information system, the structure of IS is a network composed of various components; in particular, there are three typical components, namely hardware systems, software systems and human experts who provide technological supports for the system. Similarly, node can be defined for components and arc can be defined for links among the components. It is noted that different graphs can be used to represent the IS of a service system, such as directed graph, undirected graph, weighted graph and non-weighted graph, or mixture of them. Which one is employed depends on the nature of a particular service system and a particular purpose to examine the system.

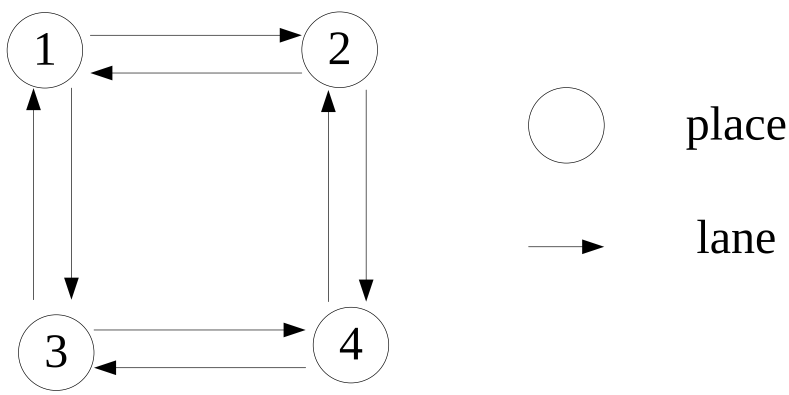

Figure 4.2 Structure of a transportation system

(2) Structure of the SS

The structure of SS refers to different types of substances, the connections among different types of substance and the distribution of different types of substance, as shown in Figure 4.3. For the enterprise information system, as mentioned above, the substance refers to enterprise information flows in the enterprise information system; in particular, the substance may be classified into different types depending on their different properties or attributes. For example, the substance of a particular enterprise information system may include different types of information flows, such as financial information, production material information, and production process information and so on. In the case of a transportation system, the substance includes different types of humans with different transportation tools and different relationships among them. The different transportation tools will enable humans to have different behaviours; instances of the different relations may be such that two or more people belong to the same family, and they had better be transported together.

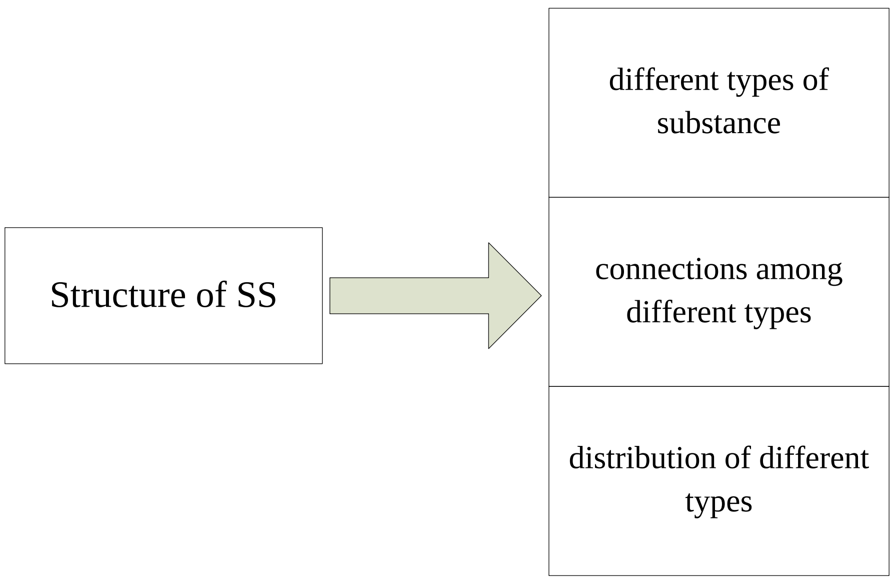

Figure 4.3 Structure of SS

At a particular time, the substance of an information system stays on a particular node or arc of the infrastructure system. Therefore, the distribution state of the substance is also considered in the representation of the structure of the SS. Again, different substances have different features. For example, the information of the enterprise information system moves very fast in the connection line. The information is usually viewed to stay on a particular node, namely a terminal or storage and ignore the transmission time when modeling. Therefore, only the distribution state of substance on the nodes can be used to represent the structure of SS for the enterprise information system. Take another typical service system, a water supply system, as an example. The substance, water, has different features. The transmission time of water in the arcs may or may not be ignored, depending on the modeling accuracy. In the latter case, the distribution of water on both the node and arc is used to represent the structure of the water supply system.

4.4.3 Behaviours and States of the IS and SS

(1) Behaviour and State of the IS

Entities in the IS are perceived by a set of properties, and these properties are called states in the FCBPSS framework. For the IS of an enterprise information system, states thus refer to the properties of the hardware systems, software systems and human experts. For example, a data storage system may have the following states: available memory space, readability, writability and so on. For the IS of a transportation system, the states refer to the properties of the roads and places, as shown in Figure 4.4, including the state of place, characterized with {initial occupancy, place capacity} (Figure 4.4), the state of road, characterized by (travel time, lane capacity) where a road has several lanes with different directions (Figure 4.4), and flow patterns.

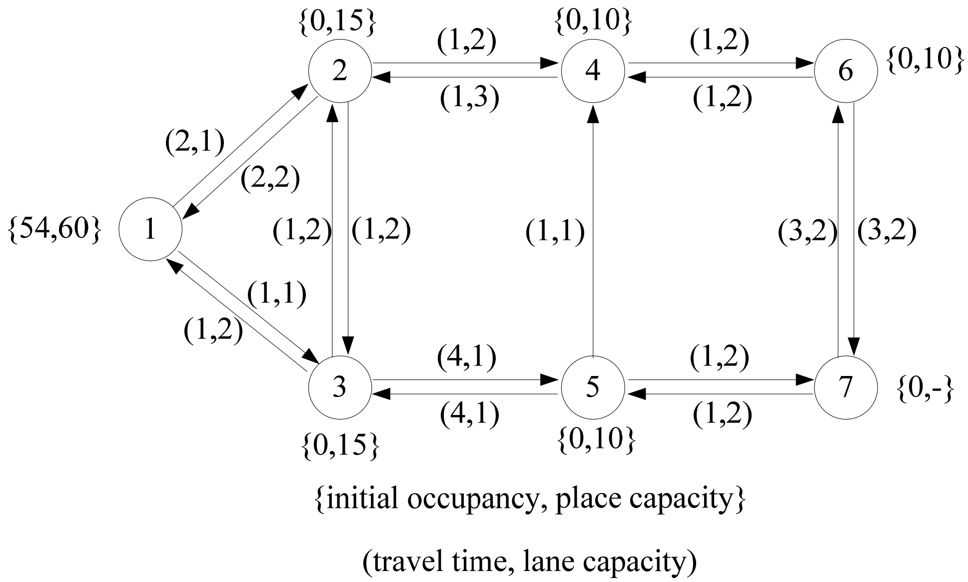

Figure 4.4 States of IS of a transportation system

The behaviour of the IS is the causal relationship among its state variables. For the IS of an enterprise information system, the behaviour of a component, say storage, may refer to the change of the available memory space, or readability or writability. In the case of a transportations system, the behaviour of IS may refers to that the road changes from one flow pattern to another flow pattern, as shown in Figure 4.5.

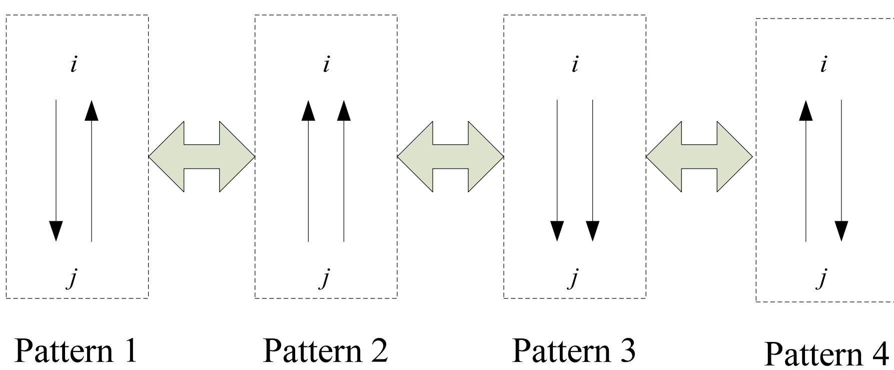

Figure 4.5 Behavior of IS of a transportation system

(2) Behaviour and State of the SS

The states of the SS refer to the properties of the substance flow on the IS. For an enterprise information system, the states of the SS, may refer to the amount of information on nodes and the rate of the information flow on edges. In the case of a transportation system, the states of the SS, as shown in Figure 4.6, include the numbers of humans and vehicles on the places, the velocities of humans and vehicles on the roads, and so on.

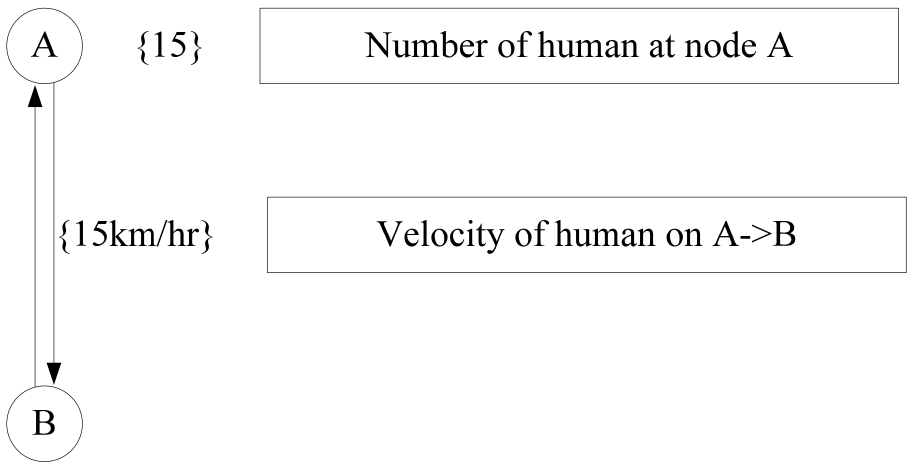

Figure 4.6 State of SS of a transportation system

The behaviour of the SS is the causal relation among its state variables. For an enterprise information system, the behaviour of the SS may refer to the change of amounts of information on nodes and rates of information flows on edges. In the case of a transportations system, the behaviour may refer to the change of the numbers of humans on different places, as shown in Figure 4.7.

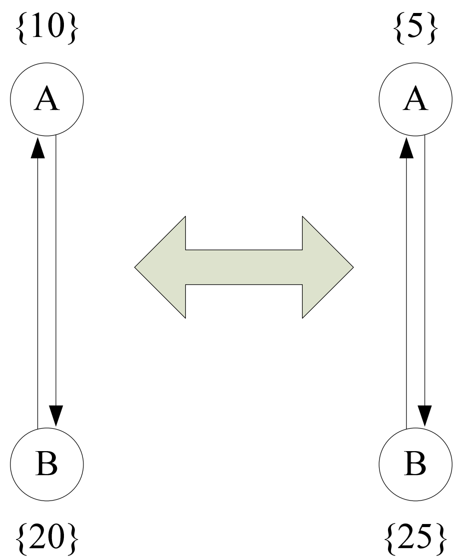

Figure 4.7 Behavior of SS of a transportation system

4.4.4 Principles of the IS and SS

(1) Principle of the IS

The principle of the IS governs the behaviour of objects or subsystems in IS. For an enterprise information system, as mentioned above, one of the behaviours of a particular storage component may refer to the change of the available memory space. However, such behaviour must obey the principle that the available memory space changes within a range between the predefined minimum bound and maximum bound. An enterprise information system is built upon a particular computer network system, the principles of IS include the protocols and controls of different objects or subsystems in IS, which decompose the whole IS into seven layers in logic, namely application layer, presentation layer, session layer, transport layer, network layer, data link layer and physical layer. For a transportation system, as mentioned in Section 4.4.3, one of the behaviours is that the road changes from one flow pattern to another flow pattern. This behaviour, as shown in Figure 4.5, must obey the following principle: the number of different flow patterns is 4 and the flow pattern of one road must change within the domain.

(2) Principle of the SS

The principles of SS govern the behaviours of SS. For the SS of an enterprise information system, the principles may refer to the flow protocols that govern the different types of information flows over the IS. In the case of a transportation system, the principles of SS may refer to the traffic rules that govern the different flows on the roads, including humans and vehicles.

4.4.5 Functions and Contexts of the IS and SS

The function is defined as "a purpose in the mind of human users and can be realized by the system (structure) owing to certain behaviours existing in the structure" [Lin and Zhang 2004]. For a service system, the services provided by the system, are the functions of the system in the FCBPSS domain. The functions of the whole service system are performed by the functions of the IS and SS. The context refers to "a particular environment where a particular system operates or works or makes sense" [He 2008]. Considering its particular features, a service system has two different contexts: (1) normal context, and (2) abnormal context, as shown in Fig. 4.8.

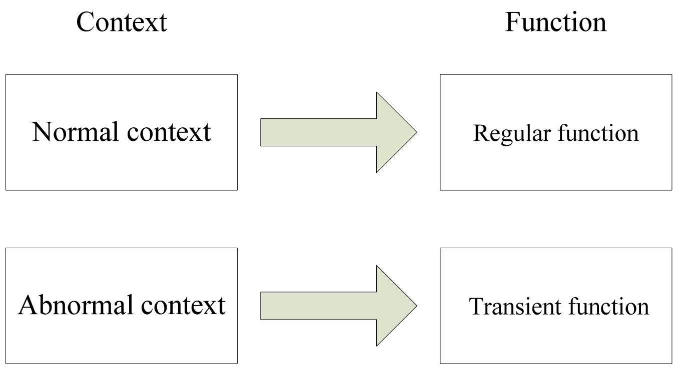

Figure 4.8 Contexts and functions of a service system

A. The normal context is a regular environment where a service system works. In this context, all components of the system are in its normal states and the functions of a service system are described as a regular function.

B. The abnormal context refers to a special circumstance where a service system works. For example, in a particular emergency situation, a part of the enterprise information system may break down. In this context, other healthy parts of the enterprise information system need to meet a larger demand than the normal situation. The functions of a service system under an abnormal context are called as the transient functions.

Next, the functions and contexts of IS and SS are discussed respectively.

(1) Function and Context of the IS

The functions of IS will be explained in the normal context and abnormal context, respectively.

In the normal context, IS of a service system has regular functions which provide regular infrastructure services to the SS. The regular function means that different components of IS work in a stable state. The regular function can be measured by the average ability of the IS to provide a stable situation for the SS. For example, the IS of an enterprise information system provides information infrastructures for different information flows in a system. In a normal situation, the IS has an average ability to support the flow of SS and further to provide information services to an enterprise.

In the abnormal context, the IS of a service system has transient functions which provide special infrastructures to the SS. In this context, the IS usually cannot work in a normal manner due to partially damages or largely increased demands, as shown in Figure 4.9. When a large-scale athletic meeting is held in a place, the wireless communication demand will increase in a huge amount. Therefore, the wireless communication system needs to meet this demand. We call the wireless communication system in this case having a transient function. Another typical example is the transportation system that needs to evacuate a large number of people from one place to another in an emergency situation. Regarding an enterprise information system, when parts of the system break down due to online attack (for example), other un-damaged parts of the system need to meet the demand of the enterprise operation. Since the transient function of the IS is to meet a special demand, it can be measured by the maximum ability of the IS to provide to the SS in an abnormal situation. To perform the maximum function, the IS may need optimal recovery solution.

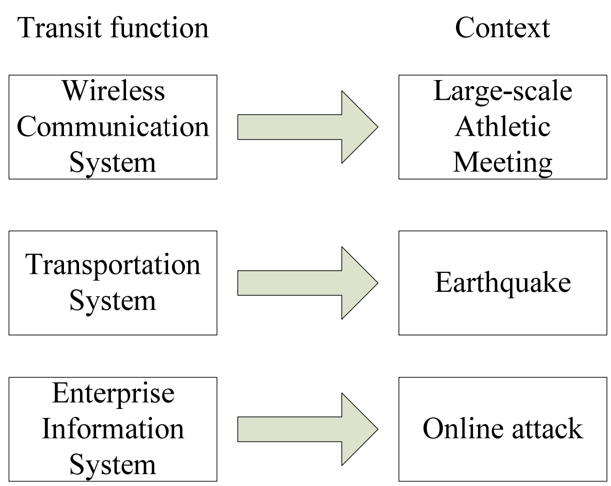

Figure 4.9 Examples of abnormal context

(2) Function and Context of the SS

In the normal context, the service of the whole service system provided to the customers is exhibited as the flow of the SS flowing on the IS. Therefore, in the normal context, SS of a service system has regular functions which provide general services to the customers of the whole service system. The regular function means different components of SS works in a stable state. The regular function can be measured by the average ability of the SS to provide for the SS in a stable situation. For example, the SS of an enterprise information system provides information services to an enterprise. This regular function can be measured by the average ability of information service provided by the SS to the enterprise.

In the abnormal context, SS of a service system has transient functions which provide special services to the customers. As discussed before, in this context, the IS usually cannot work in a normal manner due to partially damages or largely increased demands. The transient function of SS can be measured in terms of meeting a transient demand from customers in a particular abnormal situation.

4.5 Effectiveness and Usefulness of the Model

The FCBPSS approach along with its application has been shown useful in the context of various applications [Umeda et al. 1990, He 2008]. For the service system, the effectiveness and usefulness of the proposed domain model in this section are explained as follows.

4.5.1 Effectiveness of the Model

The proposed model is effective; in particular, it could play the two roles stated in Section 4.1. First, it is used as a common language for different humans involved in service systems to communicate with each other about the service system under concern. It is easy for people to have a thorough understanding of a particular service system with the proposed model. Second, the proposed framework could be further converted by a particular computer language (e.g. UML), which allows the computer system to understand the semantics of a particular service system.

4.5.2 Usefulness of the Model

First, it is useful to understand the service system and clarify a particular methodology or strategy on the system. The domain model documents the key concepts, and the domain-vocabulary of the service system. The domain model provides a structural view of the system that can be effectively used to verify and validate understanding of a problem domain among various groups of humans. It is especially helpful as a communication tool and a focusing point to understand the system and its related methodology. For example, when we undertake a management strategy for a networked service system, it will be easy to understand the strategy with the proposed domain model.

Second, it is useful to design of the service system. The system decomposition and the key concepts in the domain model give the presentation of design notion, which is the basis of the design process of the service system. In the design process, the structure of the system becomes a part of variables in the math model. The domain model provides a support to define these variables; in particular, when the topology of the system changes, the domain model provides a source of knowledge for re-defining the variables.

Third, it is useful to management of the service system. Management of the system usually involves the structure and behaviour of a system. For example, in the resilience management of the service system, the structure and behaviour of a system are necessary to be considered for creating different recovery solutions.

Fourth, it is useful to measure the properties of the service system. The key concepts in the domain model are the basis of measurement of different properties of the service system. Again take the resilience property as an example. The definitions of normal context and the abnormal context, with the regular function and transient function, not only distinguish the property domain, but also indicate a potential way to measure the resilience property of the service system from different contexts.

4.6 Variables of Service System

As an application of the domain model, variables that are responsible for the behaviour of a service system are identified.

4.6.1 Both Demands and Supplies as a Variable

As discussed in Section 3.2, the key feature of service system is that it meets human’s demand directly. This feature could be further expressed as dynamic multiple demand-supply relationships, which determine the performance of a service system.

(1) A service system has multiple demand-supply relationships. A service system may have different functions to meet human's different demands; in particular, a service system with only one function could be viewed as a special case. Furthermore, a function of a service system only makes sense for the pair of supply and demand. That is to say, for a service system, if it only manufactures goods or produces goods but no humans there as a direct receptor of the goods, the function of the service system does not make

sense. Different from service system, manufacturing systems only produce goods. Although the goods will eventually be used by humans, a manufacturing system focuses on goods production only.

(2) The demand-supply relationships are dynamic. First, the service demand changes, and the service supplier has to follow this change. A manufacturing system targets on production of goods with both known and unknown demands. When more goods are produced with a manufacturing system, the extra goods can be in an inventory. That means a manufacturing system can still move on even though there is no demand per se. This situation can never happen to a service system, and there is no sense to say service when demand is not there. So, the service system is essentially a leader-follower system. Second, a service system has many "stable" states, and they change with respect to time.

4.6.2 Network Structure as a Variable

Network structure is a variable responsible for the functioning of a service system. Service systems have a very large scale of networks. There is a paradox with the network structure. On one hand, it allows for providing more services to demands regardless of where, when and how. On the other hand, it increases the complexity of a system, which makes the system's safety more unpredictable. Zio [2009] considered that networked systems had an additional dimension of complexity in modeling of a system. As stated by Perrow in the normal accident theory [Perrow 1999], two related dimensions—interactive complexity and loose/tight coupling are the main reasons responsible for a system’s accidents. The accidents could be separated into two types; one caused by component failure and the other one caused by interaction failure [Leveson 2004b]. The aforementioned contradictory situation to the network structure can be called "network paradox". In the case of a service system, the large scale of network feature further aggravates these problems due to the presence of a large amount of components and interactions in the system but on the other hand, the network structure facilitates rapid satisfaction of demands.

4.6.3 Humans as a Variable

As a component of the service system, the human, plays two roles regarding safety performance: source of risk and source of resilience enhancement.

(1) From the traditional perspective of safety engineering, the human, as a component of the system, may cause problems for the system. The reasons could be classified into two categories: (a) human does not perform correct behaviours required by the system designer, and (b) human does perform correct behaviours; however, the correct behaviours do not handle the unexpected external disturbances, as stated by Leveson [2000], "Although many of the accidents in high-tech systems such as aircraft are being blamed on the pilots or controllers, the truth is that these systems are often designed in such a way that they are inducing new types of human behaviour and human error: The problems are not simply in the human but in the system design."

(2) From the resilience perspective, the human is viewed an essential resource for building the system's resiliency. Hollnagel proposed four abilities for building system's

resiliency, namely anticipation, monitoring, learning and responding [Hollnagel et al. 2010]. As the intelligent part of the service system, the humans are important for these abilities. Of course, the modern service system has increasingly number of artificial intelligent parts which can do similar jobs of human experts. However, the humans are still critical in the decision making.

4.6.4 Configuration of Resources as a Variable

A service system has infrastructure subsystem (IS) and substance subsystem (SS), which are considered to be resources within a service system. The configuration of the resources determines the system's ability to provide service to humans. As known, a human is a highly flexible system and the demands from humans are also flexible. The service system has to work in a flexible manner to meet such flexible demands; in particular, the flexibility is achieved through the reconfiguration of resources. A service system is a network with a large number of components, which provides the system with more options for reconfiguration.

4.6.5 Information Flow as a Variable

Information is an important substance in a service system. Information flows in two ways: (1) it flows within a service system through software and (2) it flows between a service system and its outside through internet. Thus, a service system is a software intensive and internet accessible system. This feature has critical effects on the system's performance.

The software plays an intelligent role in the service system, and it could be viewed as component or subsystem of the service system. Furthermore, it links all the other components and makes them work together. Thus, software has been a key factor in the interactions among different components. However, software may also be a source of accidents. As discussed in Section 4.6.2, there are two types of accidents, which are caused by component failure and interaction failure, respectively [Leveson 2004b]. The second type of accidents caused by interactions is usually commanded by software, which is difficult to examine with traditional safety approaches [Leveson 2004b].) The software itself may have faults which are mostly design faults due to the human error in the programming or inability to deal with some unexpected situations [Lyu 1995, Keiller 1991]. The current service systems link with the internet. A service system gets information from the internet and it may also provide service to the humans through the internet. However, it is well-known that the internet is a source of risks.

4.7 Conclusions

This chapter proposed a framework for domain modeling for service system. The novelty of the proposed domain modeling framework for the service system has been contributed by the unified definition of the service system and the domain modeling tool FCBPSS. In comparison with the related method of Umeda and Tomiyama [31], the FCBPSS framework includes a complete set of concepts underlying a system, being able to capture the semantics of why, how, where, and when the means achieves the end. The FCBPSS framework thus goes beyond the object-orientation paradigm in modeling information and knowledge. Based on the domain model, this chapter further discussed the variables of a service system, which have critical influences on its performance.

CHAPTER 5
UNDERSTANDING RESILIENCE OF SERVICE SYSTEMS

5.1 Introduction

Having a clear understanding of resilience, especially for a service system, is another key issue of this thesis. Section 2.3 and Section 2.4 have reviewed the concept of resilience in the current literature; the results have shown two problems in understanding of resilience in the current literature. The first problem is that the current definitions do not work for service systems due to the lack of consideration of the service systems' key features. The second problem is that the relationship among three properties, namely reliability, robustness and resilience, of a system in the context of safety engineering is unclear; in particular, the resilience property was sometimes mixed up with the two other properties, even in some top journals, say Science [Bongard et al. 2006]. This situation has hindered development of technology for the resilience of service systems, e.g. resilience measurement [Rosenkrantz et al. 2005, Wang and Ip 2009], as discussed in Section 2.4.4.2.

This chapter addresses the foregoing two problems through three steps: (1) to give a definition of resilience for the service systems in Section 5.2, (2) to clarify the relationship among reliability, robustness and resilience in Section 5.3, and (3) to build a framework to facilitate modeling and analysis of the resilience of a service system in Section 5.4.

5.2 Definition of Resilience for Service System

5.2.1 Definition of Safety and Related Concepts for Service System

Definition 1: ***Supply function*** is defined as the service provided by a service system, namely the output of a service system, which meets the customers' demand. A service system may have a number of different supply functions. Suppose that a service system has *m* supply functions. Then, the supply functions could be represented as: $F_i^S, 1 \le i \le m$.

Definition 2: ***Design function*** is defined as the maximum of the service that could be provided by a service system, and it is determined by the design requirement at the design stage. A service system may have a number of design functions. Suppose that a service system has *m* design functions. Then, the design functions could be represented as: $F_i^M, 1 \le i \le m$. A design function represents the maximum ability with which a service system offers a particular service; while a supply function is the actual service provided by a service system. Obviously, the relationship between the supply functions and design functions is: $F_i^S \le F_i^M, 1 \le i \le m$.

Definition 3: ***Demand function*** is defined as the demand from customers. The customers may have different demand functions. Suppose that a customer of the service system has *m* demand functions. Then, the demand functions could be represented as: $F_i^D, 1 \le i \le m$.

There are two relationships between the supply function and demand function: balance and imbalance, which are defined as follows.

Definition 4: ***Balance*** *between the supply function and the demand function* is defined as the relationship between the supply functions and demand functions such that $F_i^S = F_i^D, 1 \le i \le m$. A service system is also called in a balanced situation if the supply function and demand function are balanced. Two implications come from this definition. As stated in Section 3.2.3, the demand function of a customer and the supply function offered by a service system are interactive. For example, the design function of a service system may be much larger than the actual demand at a particular time. The interactive feature of a service system will make the system's output, namely the supply function, exactly equal to the demand function. Further, the definition implies that the balance of a service system requires that all the supply functions meet the demand functions.

Definition 5: ***Imbalance*** *between the supply function and the demand function* is defined as the relationship between the supply function and demand function such that: $\exists i, F_i^S < F_i^D, 1 \le i \le m$. A service system is also called in an imbalanced situation if the supply function and demand function are imbalanced. According to this definition, if there is one supply function that cannot meet the demand function, the service system is in an imbalanced situation. It is noted that, if a supply function is larger than the corresponding demand function, the interactive feature of the service system will decrease the supply function to meet the demand function. Thus, the imbalanced situation always corresponds to the situation where one or more supply functions are less than the demand functions.

The situations of balance and imbalance may be transferable. Thus, a service system may be rebalanced. The rebalance process is defined as follows.

Definition 6: ***Rebalance*** is a process that a service system transfers from an imbalanced situation to a balance situation.

Based on the definitions and discussions above, safety of a service system is defined as follows.

Definition 7: ***Safety*** *of a service system* is defined as the dynamic balance between the multiple supply functions and the multiple demand functions.

This definition of safety is quite different from the traditional definition of safety, as discussed in Section 2.4. According to the traditional understanding of safety, a system is not safe because some components fail as discussed in category I of safety engineering in Section 2.4.2, or because complex non-linear interactions among different coupled components and disturbances lead to emergent phenomena as discussed in category II of safety engineering in Section 2.4.3. In other words, a system is not safe because something wrong with the system. However, the proposed definition indicates that even there is nothing wrong in a system (particularly service system), it may be still not safe, as it may not meet the customers' demands. The failure of the power system in India that happened in July 2012 introduced in Chapter 1 is a good example. The investigation of this accident has shown that the initial reason of the accident was over use the power from the system.

5.2.2 Definition of Resilience for Service Systems

The **resilience** of a service system is defined as a property that allows the system to rebalance the supply function and demand function from imbalanced situations. Four remarks are given below for further explanation on this definition.

Remark 1: According to the definition of safety in Section 5.2.1, an imbalanced situation means an unsafe situation.

Remark 2: The domain model of service system in Chapter 4 has shown that a service system could be examined from different perspectives, such as state, structure, function and so on. The proposed definition of resilience implies that the resilience property should be examined from the perspective of function.

Remark 3: Resilience does not aim at returning to the original states, or structure or functions of the system; it aims at making the supply functions meet the demand functions.

Remark 4: The imbalanced situation implies that supply functions are less than demand functions. Therefore, the ability of a resilient service system is to respond to the imbalanced situations and to meet the demand functions. This point will be given further discussion in Section 5.4.

5.2.3 Comparison with Other Definitions

Section 2.4.4 reviewed three categories of definitions of the resilience concept in the context of safety engineering. A summary of the proposed definition of the resilience concept vs. other definitions are given below.

(1) Proposed definition vs. category I.

As discussed in Section 2.4.4.2, the definitions of category I use reliability or robustness to define resilience. According to the proposed safety definition in Section 5.2.1, the concepts of reliability and robustness are both related to the balance situation of a system; while, the concept of resilience is related to transferring from the imbalanced situation to the balance situation, namely rebalance process. The details of the relationship among reliability, robustness and resilience will be given further discussion in Section 5.3.

(2) Proposed definition vs. category II.

As discussed in Section 2.4.4.2, category II defines resilience as the ability of a system to recover to meet the demand from a partial damage [Zhang 2007]. This is a typical traditional understanding of the resilience concept, which considers that the imbalanced situation of a service system is caused by partial damages with the system. Therefore, category II could be viewed as a special case of the proposed definition. In other words, there may be different reasons that lead to the imbalanced situations of a service system; a partial damage is a typical reason, which decreases the supply functions. Obviously, another possible reason is the increase of demand functions. Thus, the proposed definition covers the scope of the definition of category II.

(3) Proposed definition vs. category III.

As discussed in Section 2.4.4.2, the definition of category III is a generic and all-inclusive definition which covers the traditional concepts of safety, reliability and robustness. Obviously, the definition of category III also covers the scope of the proposed definition of resilience.

The comparisons above show that: (1) the proposed definition of resilience considers the special features of the service systems, and (2) the scope of the proposed definition has overlapping area with other definitions. A brief summary is given in Table 5.1.

Table 5.1 Proposed definition of resilience VS others' definitions of resilience

Category	Key feature of Definition	relationship
I	Reliability or robustness.	Category I $\neq$ proposed definition
II	Recover ability from partial damage	Category II $\subset$ scope of proposed definition
III	All-inclusive	Category III $\supset$ scope of proposed definition
Proposed Definition	Rebalance ability from imbalanced situations	

5.3 Reliability VS Robustness VS Resilience

As discussed in Section 2.4, safety of a system has ever been described by three properties, namely reliability, robustness, and resilience. This section tries to clarify the relationships among the three Rs (i.e., Reliability, Robustness, Resilience) from different perspectives.

5.3.1 Different Understandings of Safety

Reliability refers to a traditional understanding of safety expressed in category I of safety engineering. Such an understanding, as discussed in Section 2.4.2, considers that accidents are caused by component failure, and the accidents spread as a linear chain. Reliability is a component-based property [Leveson 2011]. Safety and reliability has a close relationship and they may have the same connotations with respect to the simple system. In the development of reliability engineering, there was a main shift from component centric focus to the system level reliability [Saleh and Marais 2006]. Thus, the modern definition of reliability has been applicable to both component and system.

Robustness reflects the human's understanding of safety expressed in category II of safety engineering. Such an understanding, as discussed in Section 2.4.3, considers safety as an emergent property of the whole system, and accidents happen due to the complex non-linear interactions among different coupled components and external and internal disturbances. Thus, robustness is a property of a system not component. Robustness is a property built upon reliability. A system is viewed to be robust by meeting two conditions: (1) the system is reliable, and (2) the system is insensitive to noises.

Resilience reflects the human's understanding of safety expressed in category III of safety engineering. This understanding also considers safety as an emergent property of the whole system and accidents spread as complex interactions in the system; furthermore, safety is not about a stable state, and it is a dynamic balance between the thing that makes the system go right and the thing that makes the system go wrong [Hollnagel et al. 2011]. Resilience further considers how to increase the things that make the system go right.

5.3.2 Different Safety States

Both reliability engineering and robustness engineering try to design a safe system and maintain the system at the safe state. Thus, reliability and robustness both consider that a system has two states only: failure and functioning.

However, resilience considers that safety and failure are not only two different states. There may be multiple states between safety and failure; furthermore, there may be multiple safe states and multiple unsafe states. For example, a service system may have different balanced situations which correspond to different supply and different demand; it may also have different imbalanced situations.

It has been discussed in Section 5.2 that the imbalanced situations may be caused by partial damage of a system. It is noted that the system’s reliability property also involves the situation that a system has some damaged component(s); however, the reliability approach only uses redundancy to solve this issue. As discussed in Section 2.3.2, the assumption for the repairable reliability is that the time to repair is assumed to be negligible and the system is in the continuous operation without interruption [Birolini 2010]. In other words, a system may have a damaged component; but, this system is only viewed to be reliable when it has a redundant component or its function could be implemented by other components. However, in resilience engineering, recovery of the damaged component(s) or subsystem(s) is considered without assumption of any continuous operation. Therefore, one key issue in resilience engineering is to know what to do; in particular, how to make a system respond to the imbalanced situations where a part or parts of the system are damaged.

5.3.3 Different Focuses

The assumption of reliability of a system relies on the reliability of all of its components. Thus, the reliability property focuses on whether each component of a system can work as designed. According to the domain model of service system proposed in Chapter 3, it can be seen that reliability actually examines a system from a structural perspective. When dealing with a recovery situation, the system needs to keep its functions even during recovery process. Therefore, continuous operation is important for the reliability property; furthermore, the repairable ability only applies to the redundant items [Birolini 2010].

The robustness of a system focuses on a system's performance under different noises; in particular, a system needs to maintain its functions against internal and external noises. According to the domain model of service system proposed in Chapter 3, it can be seen that robustness actually examines a system from a functional perspective. The means to improve robustness is the component or subsystem design [Taguchi 1986] and operation management.

The resilience of a system focuses on a system's performance facing damage or other unsafe situations; in particular, the system can return to the safety state from the unsafe state. Therefore, the recovery ability or the rebalance ability in the context of service system, is necessary for the system's resiliency. The reliability property may also deal with the recovery, as discussed in Section 2.4.2; however, its recovery ability only applies to the failed redundant parts. Thus, from the reliability perspective, the time for recovery is assumed to be negligible and the system will not experience any break during recovery process. The recovery ability in resilience is beyond the scope of (component) redundancy; furthermore, resilience depends on the system's ability to reconfigure all resources. This reconfiguration ability is not required by reliable or robust systems.

5.3.4 Being Coupled

The three Rs are coupled, which means that there are some overlapping areas in the three concepts. There are two perspectives to analyze these coupled relationships. The first perspective is on the development of each concept. The second perspective is in the context of a particular system, say a service system. The coupled relationships among three Rs are discussed from these two perspectives in the following sections.

5.3.4.1 Perspective of concept development

The developments of the three R concepts in the field of safety engineering have some similar experience, which may be a general phenomenon in any knowledge development. That is, a concept when it was originally proposed usually has a relatively narrow scope; however, with the development of the concept, its scope trends to be larger. The three Rs are good examples to reflect this phenomenon.

When originally proposed, the basic scopes of the 3Rs are as follows. (1) the reliability concept mainly considers if the system could remain its normal functions and does not care about the noise, (2) the robustness concept only considers noise and its goal is to make the system remain its required functions under different noises, and (3) the resilience concept mainly considers how to rebalance the system from some imbalanced situations.

However, with the further developments, all the three concepts were extended to the more general meanings.

(1) Extension of reliability. The reliability concept in some current literature also deals with a system's performance under noises. For example, as discussed in Section 2.4.2.3, Billinton and Allan [1996] decomposed the reliability of the system into two concepts,

adequacy and security. They further defined system security as "the ability of a system to respond to disturbances arising within that system" in [Billinton and Allan 1996]. Their work of reliability has already involved the scope of robustness. The difference is that the robustness concept considers not only the internal disturbances but also the external disturbances. Reliability has also been extended to the scope of resilience in terms of transportation system [Victoria Transport Policy Institute 2010].

(2) Extension of robustness. Bongard et al. [2006] considered robustness as the same with resilience, which means that the robustness property also deals with the situation of system's partial damages. Although, as discussed in the beginning of this chapter, the research in this thesis does not agree with such an extension, this is a trend of concept extension for robustness in this case. It is noted that this is a paper published in Science, which always has a reputable name. Little [2003] viewed the robustness of a system as a combined property of reliability and resilience.

(3) Extension of resilience. As discussed in Section 2.4.4, resilience recently has been extended to the whole scope of system's safety; in particular, it is viewed as the ability to succeed under varying conditions [Hollnagel et al. 2010]. Therefore, the resilience concept has covered the scopes of reliability and robustness in the recent literature.

5.3.4.2 Perspective of service system

From the perspective of service system, as defined in Section 5.2, resilience refers to the ability of rebalancing from imbalanced situations. Similarly, reliability and robustness refer to the abilities of keeping a system in the balance situations; in particular, robustness focuses on the system's performance under noises. The balance situations mean that the supply functions meet the demand functions. However, in practice, there must be a bound that limits the difference between the supply function and demand function; a difference beyond the bound will lead to an imbalanced situation. In other words, there is not an absolute broader line between the balanced situation and the imbalanced situation; there is a vague area between the balanced situation and imbalanced situation. Therefore, there is also a vague area between the properties of reliability, robustness and resilience.

5.3.5 The Three Rs May Have Conflicts

As the three Rs focus on different aspects of system's safety performance, they may have some conflicts with different approaches to achieve their goals.

(1) High reliability may result in less robustness and resilience. The important approach for reliability is redundancy. Designer must decide a proper redundancy degree for critical elements; thus, the costs of redundant systems will be high [Anderson and Bartholdi 2000]. The reconfiguration ability is an important factor related with a system's resilience. The management of information of such a system will be difficult and costly especially when the redundant elements need to be synchronized from time to time. For example, the replication of a large database takes a long time and also uses a large network bandwidth. Increased redundancy will also increase the inconsistency between elements [Liu et al. 2010]. The design and control of such a system will be more

complicated, which will affect the system's robustness and resilience. Furthermore, the limited resource is an important assumption of reliability [Birolini 2010]; therefore, high cost for redundancy will affect a system's effort on improvement of robustness and resilience.

(2) High resilience may result in less reliability and less robustness. Liu et al. [2010] explained that resilience may have conflict with other system attributes. The means to achieve resilience is adjusting system's resources and reconfiguration of its structure. Furthermore, the reconfiguration of structure is based on the modularization. However, these two concepts will increase the number of interfaces where errors are prone to occur, which is the important source of system's uncertainty. Therefore, resilience enhancement may result in less reliability and less robustness.

(3) High reliability and robustness may result in less resilience. Similar with the second point above, a reduction in the number of interfaces is important for improvement of reliability and robustness. However, this may reduce the possible reconfiguration options for a system and reduce the adjusting ability of the system. Therefore, high reliability and robustness may affect a system's resilience.

5.4 A Framework for Resilience Analysis

Upon the discussion on the concepts of service systems and resilience, this section establishes a framework with which to analyze the resilience property of the service systems.

5.4.1 Framework

There are two basic issues of resilience research on the service systems, namely measurement and improvement, as discussed in Chapter 1. This section presents a framework for resilience analysis on a service system, which facilitates further developments on the methodology of these two issues in Chapter 6 and Chapter 7, respectively.

According to the definition proposed in Section 5.2, resilience of a service system can be measured through its ability of rebalancing the supply functions and demand functions from imbalanced situations. The rebalance ability of a service system means that the system can generate some rebalancing solutions under the imbalanced situations. Obviously, the available "best" rebalancing solution will determine the resilience performance of a service system. The improvement of the resilience performance of a service system refers to generating a better rebalancing solution. Thus, the two issues, measurement and improvement, are coupled, as they are both related with the rebalancing solutions. The differences between the two issues are: (1) the measurement issue considers to how to evaluate or define the current available "best" rebalancing solution, and (2) the improvement issue considers how to generate a better rebalancing solution for a service system.

According to the definition of rebalance, there may be two possible ways to eliminate the difference between the supply functions and the demand functions: (1) to increase the supply functions and (2) to decrease demand functions. The increase in supply functions further depends on the available resources of a service system and how to use the resources, namely reconfiguration of the resources. The decrease in demand functions usually has two options: (a) to increase the price of service, and (b) to cut some demand functions directly. The first option is a flexible way to decrease the demand from an economic perspective. The second option is a mandatory way under some emergency situations. For example, in the evacuation process, some roads may be constrained to be one-directional.

Therefore, the resilience analysis on a service system could be summarized as Figure 5.1.

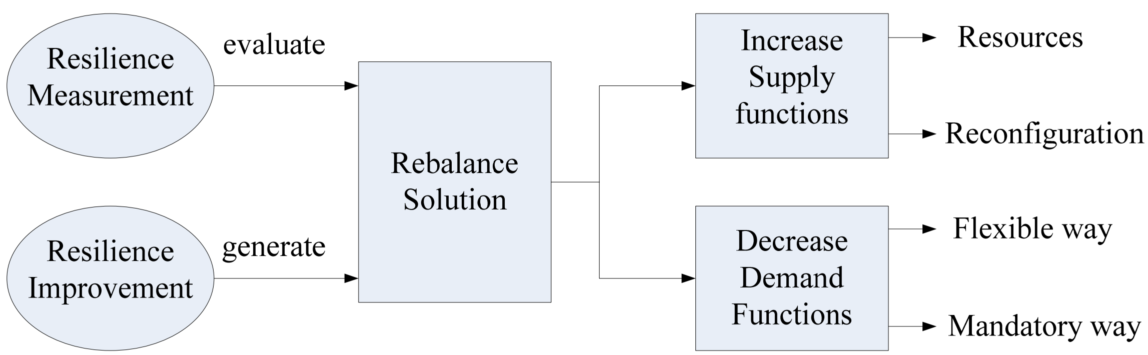

Figure 5.1 Resilience analysis on a service system

Several remarks are made below as s summary of the resilience analysis framework presented in Figure 5.1.

Remark 1: There are two basic issues of resilience analysis on a service system, measurement and improvement.

Remark 2: Both measurement and improvement of resilience are related to rebalancing solutions, which could be generated in two ways: increase of supply functions and decrease of demand functions.

Remark 3: Decrease of demand functions could be implemented through flexible ways or mandatory ways. The corresponding methods, however, are not generated from an engineering perspective.

Remark 4: Increase of supply functions depends on the available resources and reconfiguration of the resources. The corresponding methods are generated from an engineering perspective, which is the focus of this thesis.

Next section will discuss the available resources of a service system for its resilience.

5.4.2 Available Resources for Resilience

The conceptual model proposed in Chapter 4 has pointed out that there are two main categories of resources within a service system, namely infrastructure system and substance system. There may be some extra input resources to a service system for development of the rebalancing solutions. Therefore, the available resources for resilience analysis on a service system could be viewed as two categories, as shown in Figure 5.2: (1) inherent resources, including IS and SS, and (2) extra input resources. If a service system rebalances itself only through the inherent resources, it could be viewed as a self-healing system.

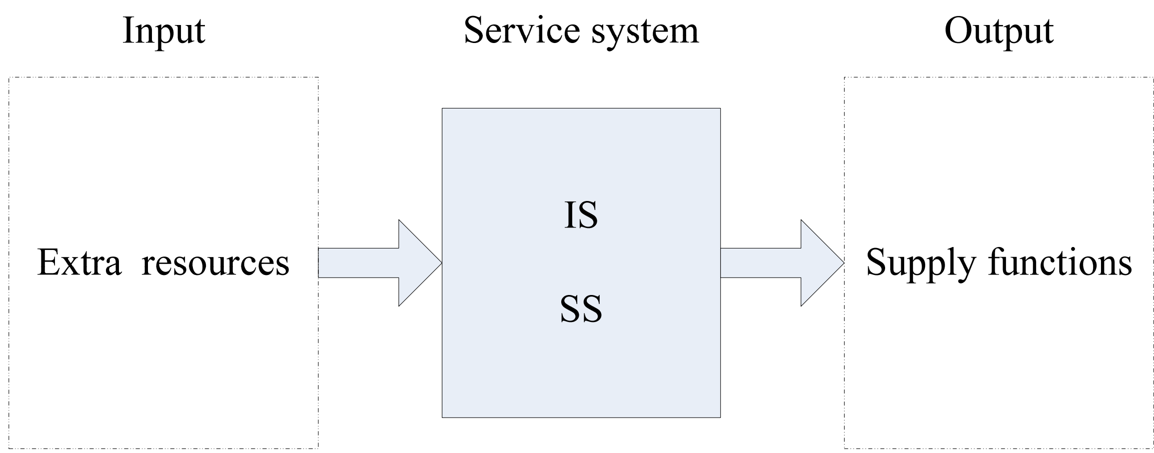

Figure 5.2 Resources for resilience of a service system

The definition of service system in Section 3.3 and the definition of balance in Section 5.3.1 both discussed the interactive feature between the supply functions and demand functions; in particular, the demand functions affect the supply functions. Thus, the demand functions could also be viewed as a kind of input, namely input information, to the service systems.

5.5 Conclusions

(1) Safety and resilience are defined in terms of a service system. The definitions are proposed by considering the important features of service systems.

(2) The relationships among reliability, robustness and resilience are clarified. The three Rs reflect human's different understanding of safety and have different focuses. They are coupled concepts and may also have conflicts.

(3) A framework is proposed to analyze the resilience property of a service system. This framework considers two foundational issues of resilience analysis, measurement and improvement; furthermore, both of them rely on the rebalancing solutions under imbalanced situations. The rebalancing solutions rely on available resources and their reconfiguration strategy.

CHAPTER 6
MEASUREMENT OF RESILIENCE

6.1 Introduction

This chapter presents a methodology of the resilience measurement for a service system. First, axioms of resilience measurement are discussed in Section 6.2. Second, mathematical models based on these axioms are developed in Section 6.3. Third, two cases, transportation system and enterprise information system, are employed to illustrate and validate the proposed measure in Section 6.4 and Section 6.5, respectively. At last, conclusions are given in Section 6.6.

6.2 Axioms for Resilience Measurement

Following the discussion in Chapter 5, this thesis measures the resilience of a service system by maximization of the set of imbalanced situations, which can be rebalanced with bounded time and resources. Four important corollaries are derived in the following, which are taken as the axioms for the resilience measurement.

Axiom 1: The resilience of a system can only be measured for a particular imbalanced situation.

Axiom 2: A system which can rebalance the supply and demand relation from a larger imbalanced situation is more resilient.

Axiom 3: A system which can rebalance the supply and demand relation from the imbalanced situation with less time is more resilient.

Axiom 4: A system which can rebalance the supply and demand relation from the imbalanced situation with less resource is more resilient.

The above axioms show that there are three factors affecting a rebalancing solution: (1) imbalanced situation, (2) rebalancing time, and (3) rebalancing cost. It is noted that Allenby and Fink [2005] had a similar thought with axiom 1; in particular, they claimed that "resiliency is not a global characteristic of a system and it can be meaningfully determined only with reference to an identified system and particular challenges" [Allenby and Fink 2005]. Axiom 1 implies that a service system may be (1) resilient under an imbalanced situation say situation A, and (2) not resilient under another imbalanced situation say situation B. Thus, it is reasonable to measure a service system's resiliency in terms of a particular situation. It is not reasonable to compare two different systems under different imbalanced situations.

It is noted that axiom 2 is similar with the measurement in robust control, which finds the bound on perturbations that the system can endure and such a bound is viewed as the measure of robustness [Cheres et al. 1989, Chen and Han 1994, Feng and Brandt 1998]. The difference lies in that the robust control method requires a system to be in a stable or reliable condition; while a system's resilience may consider non-stable or imbalanced situations.

For a networked service system under a particular imbalanced situation, there may be different rebalancing solutions and the best solution will determine the resilience of the system. The rebalancing solutions certainly depend on the available rebalancing time and resources. The axioms actually imply that given different conditions, there may be different measurements. For example, given bounded rebalancing time and resources, the measurement is a maximization of imbalanced situations. Given an imbalanced situation and available rebalancing cost, the measurement is a minimization of rebalancing time. Next, mathematical models based on the foregoing axioms are presented, respectively.

6.3 Mathematical Models for Resilience Measurement

This section presents mathematical models based on the foregoing axioms, including assumptions, variable definitions, objective functions and constraints. Simple examples are also given to illustrate the models.

6.3.1 Model based on Axiom 2

6.3.1.1 Assumptions

(1) Given required rebalancing time for each function and total resource constraints.

(2) All the supply functions and demand functions are rebalanced within the corresponding required time.

6.3.1.2 Variable definition

Table 6.1 Variable definition in the model based on Axiom 2

m	: the total number of functions;
n	: the number of categories of resources;
w_i	: weight of function i, $i = 1, 2, \cdots, m$, $\sum_{i=1}^{m} w_i = 1$;
$r_{i,j}^{t}(x)$	: the number of resource j needed by function i at time t with rebalancing solution x, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$;
R_j	: the total number of resource j, $j = 1, 2, \cdots, n$;
D_i^t	: the demand function i at time t, $i = 1, 2, \cdots, m$;
$S_i^t(x)$	: the supply function i at time t with rebalancing solution x, $i = 1, 2, \cdots, m$;
T_i	: the required time for the rebalance of function i, $i = 1, 2, \cdots, m$;
x	: rebalancing solution

Note that rebalancing solution x is the decision variable of the model.

6.3.1.3 Objective function and constraints

The objective is to maximize the imbalanced situation that can be rebalanced, as shown in Fig. 6.1. In Fig. 6.1, the upper solid strait line is the demand function i (D_i^t), which is supposed to be a constant; the dashed curve is the supply function with a rebalancing solution x. The lower solid straight line labels the initial supply function at time 0 without any rebalancing solutions. Within the given required time, the supply function with rebalancing solution must meet the demand function.

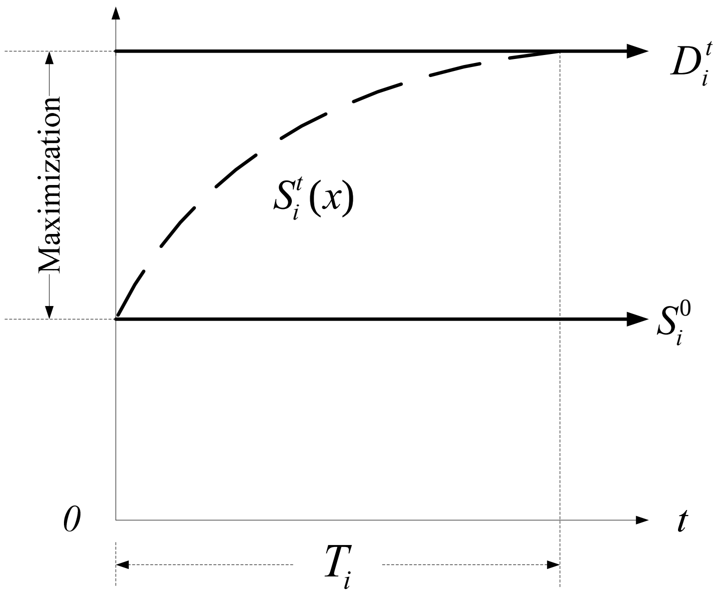

Figure 6.1 Maximization of imbalanced situation

$$\text{Max} \quad \sum_{i=1}^{m} w_i \frac{D_i^0 - S_i^0(x)}{D_i^0} \tag{6.1}$$

$$\text{s.t.} \quad S_i^t(x) \geq D_i^t, t \geq T_i,\ i = 1, 2, \cdots, m \tag{6.2}$$

$$\sum_{i=1}^{m} r_{i,j}^t(x) \leq R_j,\ j = 1, 2, \cdots, n, t \leq \max\left\{T_i \middle| i = 1, 2, \cdots, m\right\} \tag{6.3}$$

In the above, Eq. (6.1) represents the imbalanced situation at beginning; Eq. (6.2) represents the constraints of rebalancing time for different functions; Eq. (6.3) represents the resource constraints.

6.3.1.4 Illustrtion

A very simple transportation system example is given to show how the model works. In this example, we only consider the maximum imbalanced situation for which the system can rebalance and the rebalancing time and cost are ignored. Fig. 6.2 is an original transportation system with only two nodes. Two edges link the two nodes. The travel time and edge capacity are as shown in the figure. Suppose that the unit of travel time is minute. In this example, the imbalanced situation is assumed such that the transportation demand from A to B increases from 10 per 2 minutes to 15 per 2 minutes due to some reason. The rebalancing solution is based on contraflow approach [Wolshon et al. 2005], namely reversing the edge B to A, as shown in Fig. 6.3. With this rebalancing solution, the maximum transportation ability from A to B becomes 20 per 2 minutes. Therefore, the largest imbalanced situation that can be rebalanced is such that the transportation demand from A to B rises to 20 and the imbalance degree is (20-10)/20=50%. Thus, the resilience of this transportation system facing the particular imbalanced situation due to the increased transportation demand from A to B is 50%.

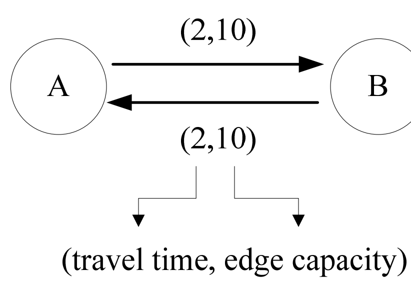

Figure 6.2 Original transportation system with only two nodes

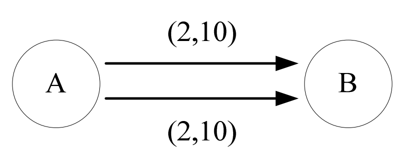

Figure 6.3 Rebalancing solution

6.3.2 Model based on Axiom 3

6.3.2.1 Assumptions

(1) Given the imbalanced situation, required rebalancing time and resource constraints.

(2) All the supply and demand functions are rebalanced within the corresponding required time.

It is noted that axiom 3 implies that the actual rebalancing time does not equal to the required rebalancing time.

6.3.2.2 Variable definition

Table 6.2 Variable definition in the model based on Axiom 3

M	: the total number of functions;
N	: the number of categories of resources;
w_i	: weight of function *i*, $i=1,2,\cdots,m$, $\sum_{i=1}^{m} w_i = 1$;
$r_{i,j}^{t}(x)$	: the number of resource j needed by function *i* at time *t* with rebalancing solution *x*, $i=1,2,\cdots,m$, $j=1,2,\cdots,n$;
R_j	: the total number of resource *j*, $j=1,2,\cdots,n$;
$C_i(x)$	: the completion time for the rebalance of function *i*, $i=1,2,\cdots,m$;
D_i^t	: the demand function *i* at time *t*, $i=1,2,\cdots,m$;
$S_i^t(x)$	: the supply function *i* at time *t* with rebalancing solution *x*, $i=1,2,\cdots,m$;
T_i	: the required time for the rebalance of function *i*, $i=1,2,\cdots,m$;
X	: rebalancing solution

Note that rebalancing solution *x* is the decision variable of the model.

6.3.2.3 Objective function and constraints

$$\text{Min} \quad \sum_{i=1}^{m} w_i \frac{C_i(x)}{T_i} \tag{6.4}$$

$$\text{s.t.} \quad S_i^t(x) \geq D_i^t, t \geq C_i(x), i = 1, 2, \cdots, m \tag{6.5}$$

$$\sum_{i=1}^{m} r_{i,j}^t(x) \leq R_j, j = 1, 2, \cdots, n, t \leq \max\left\{T_i \middle| i = 1, 2, \cdots, m\right\} \tag{6.6}$$

$$C_i(x) \leq T_i, i = 1, 2, \cdots, m \tag{6.7}$$

In the above, Eq. (6.4) represents the resilience of the system; Eq. (6.5) represents the actual rebalancing time for each function; Eq. (6.6) represents the resource constraints; Eq. (6.7) represents the constraint of rebalancing time.

6.3.2.4 Illustration

A simple example of enterprise information system is employed to show how the model works. An enterprise information system is a very special service system in that such a system usually has backup and could be rebalanced from even 100% lost of functions. We consider a scenario that an enterprise information system is fully damaged. There are two functions in this system, which are totally lost. There are two categories of resources. The total number of resource 1 is 2; the total number of resource 2 is 4. All the functions are treated with the same importance and thus the weight for each function is 0.5. Due to the limitation of resources, the two functions cannot be rebalanced synchronously. This implies that a rebalancing solution needs to choose a particular order among a set of rebalance tasks. The information of rebalancing solutions is given in Table 6.3. p_i is the process time for the recovery of function *i*. Obviously, the optimum solution for this problem is with the order of [1,2]. The fitness value is 0.9375, which implies that the system could be rebalanced a little earlier than the required recovery time.

Table 6.3 Rebalance parameters

Function	Resource 1, $r_{i,1}$	Resource 2, $r_{i,2}$	p_i (min)	T_i (min)
i=1	2	3	2	2
i=2	2	4	5	8

6.3.3 Model based on Axiom 4

6.3.3.1 Assumptions

(1) Given the imbalanced situation, required rebalancing time, and resources constraints.

(2) All the supply and demand functions are rebalanced within the corresponding required time.

Axiom 4 implies that the actual used resources do not equal to the resource limits.

6.3.3.2 Variable definition

Note that rebalancing solution x is the decision variable of the model.

Table 6.4 Variable definition in the model based on Axiom 4

m	: the total number of functions;
N	: the number of categories of resources;
w_i	: weight of resource j, $j = 1, 2, \cdots, n$, $\sum_{i=1}^{n} w_j = 1$;
$r_{i,j}^{t}(x)$	: the number of resource j needed by function i at time t with rebalancing solution x, $i = 1, 2, \cdots, m$, $j = 1, 2, \cdots, n$;
R_j	: the total number of resource j, $j = 1, 2, \cdots, n$;
$U_j(x)$	: the required resource j with rebalancing solution x, $j = 1, 2, \cdots, n$;
T_i	: the required time for the rebalance of function i, $i = 1, 2, \cdots, m$;
D_i^t	: the demand function i at time t, $i = 1, 2, \cdots, m$;
$S_i^t(x)$	: the supply function i at time t with rebalancing solution x, $i = 1, 2, \cdots, m$;
x	: rebalancing solution

6.3.3.3 Objective function and constraints

$$\text{Min} \quad \sum_{j=1}^{n} w_i \frac{U_j(x)}{R_j} \tag{6.8}$$

$$\text{s.t.} \quad S_i^t(x) \geq D_i^t, t \geq T_i \tag{6.9}$$

$$\sum_{i=1}^{m} r_{i,j}^{t}(x) \leq U_j(x), j = 1, 2, \cdots, n, t \leq \max\left\{T_i \middle| i = 1, 2, \cdots, m\right\} \tag{6.10}$$

$$U_j(x) \leq R_j, j = 1, 2, \cdots, n \tag{6.11}$$

In the above, Eq. (6.8) represents the resilience of the system; Eq. (6.8) represents the constraint of the rebalancing time; Eq. (6.10) represents the actual used resources in the rebalance process; Eq. (6.11) represents the resource constraints.

6.3.3.4 Illustration

We consider a scenario that an enterprise information system is fully damaged. There are two functions in this system, which are totally lost. There are two categories of resources.

The total number of resource 1 is 4; the total number of resource 2 is 8. All the resources are treated with the same importance and the weight for each category of resources is 0.5. The information of the rebalancing solutions is given in Table 6.5. p_i is the process time for the recovery of function *i*. Due to the constraint of required rebalancing time, there are two rebalancing solutions: (1) rebalance the two functions with the order of [1,2], and (2) rebalance the two functions concurrently. Obviously, the optimum solution for this problem is the first one with the order of [1,2]. The fitness value is 0.5, which implies that the system could be rebalanced with only half of the available resources.

Table 6.5 Rebalance parameters

Function	Resource 1, $r_{i,1}$	Resource 2, $r_{i,2}$	p_i (min)	T_i (min)
i=1	2	3	2	4
i=2	2	4	5	8

6.3.4 Discussion

Several remarks are given below for further explaining of the models.

Remark 1: The models in this section are general expressions of different axioms. When the models are applied to a particular service system, the rebalancing solution x needs to be expressed in detail according to the available solution of the system.

Remark 2: The assumptions are important, which determine the context of resilience measurement. It is only with the same assumption that the resilience of two similar service systems, for example, two transportations systems, can be compared.

Remark 3: Different axioms may have conflicts. For example, to rebalance a system from a larger imbalanced situation may require more time and resources; a rebalancing solution with less time may require more resources.

Remark 4: The three models may be considered together for a service system. In this way, the measurement of resilience turns into a multi-objective optimization problem. Preferences of a service system may be given to different objectives.

6.4 Validation Case 1: Transportation System

This section applies the proposed methodology to measure the resilience of a transportation system.

6.4.1 Transportation System

As discussed in Chapter 4, a transportation system is viewed as having two levels: infrastructure level and substance level; in particular the substance level includes the substance which “flows” over the infrastructure. We view them as the subsystems of an entire transportation system, namely transportation infrastructure (TI) system and transportation substance (TS) system. They are, however, related to each other by the constraint that the TS system depends on the TI system or the TS must “flow” within the TI system. The flow of TS follows certain constraints which are called “traffic rules”. The dynamics of the transportation system is determined by the flow of TS under the constraints of these rules.

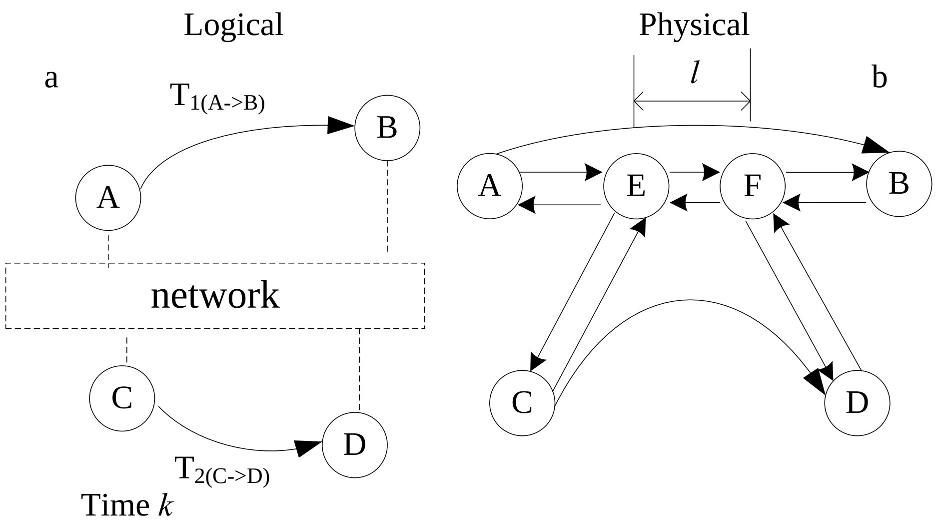

Figure 6.4 A transportation system

A transportation system is a service system that has the notions of demand and supply. The **demand** can be generally defined as moving a part of $TS_i\left(i=1,2,\cdots,n;n: the\ \ number\ \ of\ \ TS_i\right)$ from one place (*A*) to another place (*B*) within a time denoted by $T_{i(A \to B)}$ (see Fig. 6.4a). Places A and B are a part of a transportation system, especially a part of its TI system. Further, at one time say *t*, the demand may be more than one (see Fig. 6.4a where there is a demand from place C to place D at time *t* along with the demand from place A to B). In Fig. 6.4a, we show a logical level of the transportation system at time *t*, and in Fig. 6.4b, we show a possible scenario at the physical infrastructure level where the two transportation tasks (A to D; A to B) may share one segment (E to F) in the transportation system. Such a sharing essentially leads that planning of the movement of substances becomes extremely difficult. The **supply** of service is made with the substance such as vehicles and infrastructures such as roads.

6.4.2 Problem Definition

We assume that a transportation system is in an imbalanced situation and a rebalancing solution is adopted to rebalance the system. Next, the imbalanced situation and the rebalancing solution are expressed in detail.

6.4.2.1 Imbalanced situation

The scenario is that one place is in a dangerous area (single source) where victims are to be evacuated within a required time, and the other place is in a safe area (single destination) to which the victims are to be sent. A transportation network lies in between the source and destination; in particular, the transportation system is partially damaged in an emergency condition. The demand of evacuating the victims is much larger than the regular transportation demand between the source and destination. Therefore, the transportation system functioning with a regular manner cannot meet the demand, which leads to an imbalanced situation of the system.

The network system is composed of a set of places and a set of roads (Fig. 6.5). Each road has lanes which have opposite directions and are grouped in terms of their directions – i.e., there are two groups. In Fig. 6.5, edge 1->2 is a group of lanes with the same direction in road 1-2 and edge 2->1 is the other group of lanes with the opposite direction to edge 1->2 on road 1-2. We further consider the case that a road has lanes which have all in one direction as a special case of the general case that a road has lanes that have the opposite directions. The direction of the lanes of a road is defined as **flow pattern** of the road. The flow pattern of all roads of a transportation system is defined as the **flow pattern** of the transportation system. Each lane has a capacity and a travel time. The capacity of a lane is defined as the maximum number of evacuees who can be transported per unit period. Each edge also has a capacity and a travel time. The capacity of an edge is the sum of all lanes in this edge. The travel time of an edge is equal to the travel time of each lane in this edge.

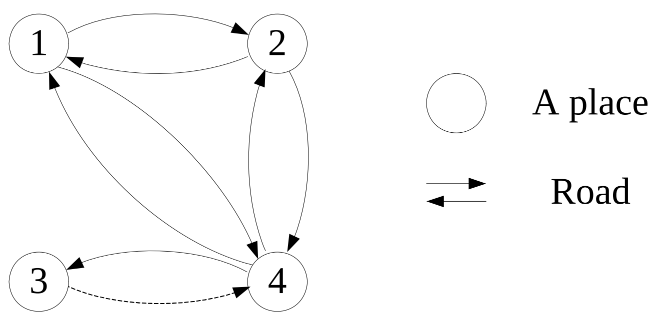

Figure 6.5 Conceptual model of the transportation system

6.4.2.2 Rebalancing solution

The regular **rebalancing solution** for the imbalance situation described in Section 6.4.2.1 is planning the substance of the system. The **planning** of a transportation system is meant: (1) to determine the flow pattern of the system, which is called task 1, and (2) to determine how the substance flows over the infrastructure with a particular flow pattern, which is called task 2. A transportation system has an original flow pattern; in particular, one road usually has two edges (or two groups of lanes) with opposite directions. Thus, task 1 of planning refers to reversing the directions of some edges, i.e., the contraflow method [Wolshon et al. 2005]. Task 1 is also called **flow pattern planning** in this thesis.

6.4.2.3 Measurement of resilience

In this problem, the transportation system needs to evacuate victims from a dangerous place to a safe place within a required evacuation time. Thus, axiom 2 and its corresponding mathematical model should be applied to measure the resilience of the transportation system; in particular, the resilience is described with maximization of the number of victims that could be evacuated within the required evacuation time.

6.4.3 Mathematical Model

The state of a road is represented in terms of edges as shown in Fig. 6.6. In this figure, the d_1, d_2, d_3, d_4 represent four states with corresponding edge patterns associated with each of them. Fig. 6.7 is the result by adding information of place (i.e., *i*, *j*) to Fig. 6.6. Variable x_{ij} is introduced to represent the information of edge *i*->*j*; in particular $x_{ij} = 1$ represents the flow from *i* to *j* and $x_{ij} = 0$ represents the flow from *j* to *i*. As such, the state of a road connecting place *i* to place *j* can be expressed by $\left(x_{ij}, x_{ji}\right)$, as shown in Fig. 6.7. It should be noted that when the travel times and capacities of two edges in a road are different, pattern d_1 and pattern d_4 are different. Only in the special case that the travel times and capacities of the two edges in one road are the same, the pattern d_1 and the pattern d_4 have the same transportation ability. This has implied the need of the consideration of the 4 flow patterns (instead of 3 patterns considered by Kim et al. [2008]) in an emergency situation. For the evacuation of victims from a source place *s* to a destination place *d*, the flow pattern of the roads associated with these two nodes should be ones as shown in Fig. 6.8. In this figure, we show that for the source place, all flows must be outgoing, and for the destination node, all flow must be incoming.

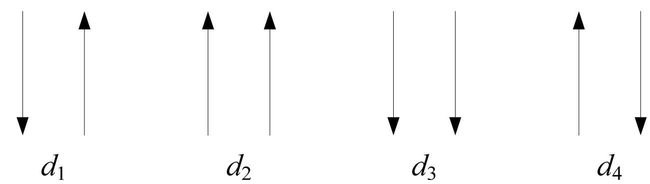

Figure 6.6 Flow patterns on a single road

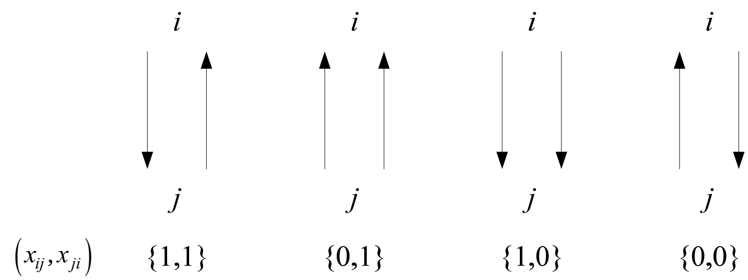

Figure 6.7 Definition of the domain of the variable to represent the flow patterns

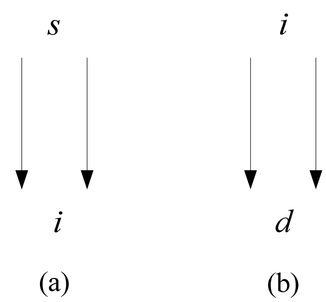

Figure 6.8 The flow patterns which are associated with source place and destination place

Let us use ‘node’ for place and ‘arc’ for edge. A particular transportation system has m nodes and n directed arcs. As such, a transportation system can be represented as a directed graph denoted by G and $G = (N, A)$. A flow pattern is, thus, a snapshot of the network G starting from the transportation or evacuation task to the completion of the task. Each node has a capacity and an initial occupancy. Each arc also has a capacity and a constant travel time. Next, the concepts about evacuation time are defined.

The **individual evacuation time** of one evacuee (ET_i , where i stands for this evacuee) is defined as a period of time from the moment when the first evacuee in the crowd leaves from a source node to the moment when evacuee i arrives at a destination node. The **total evacuation time** of all evacuees (TET) is defined as a period of time from the moment when the first evacuee in the crowd leaves from a source node to the moment when the last evacuee arrives at a destination node. The **average evacuation time** (AET) is defined as

$$AET = \sum_{i=1}^{q} ET_i \Big/ q \tag{6.12}$$

where q is the total number of evacuees in a source node. Assume that at the beginning, there are evacuees in a certain node.

The objective of resilience measurement is to find a rebalancing solution to maximize the number of victims within a given required evacuation time. It is noted that the required evacuation time refers to the total evacuation time (TET).

6.4.3.1 Variable definition

The variables are defined in Table 6.6.

Table 6.6 Variable definition in measurement model for transportation system

$G=(N,A)$	: directed network with N set of nodes and A set of arcs (static network);
$m=\lvert N\rvert$	: number of elements in set N;
$n=\lvert A\rvert$	: number of elements in set A;
c_{ij}	: capacity of arc (i,j);
λ_{ij}	: travel time of arc $(i,j), \forall (i,j)\in A$;
a_i	: node capacity (the maximum number of evacuees who can stay in node i);
S	: source node;
Q	: initial number of evacuees in node s;
$pred(i)$	: $=\{j \mid (j,i)\in A\}$, predecessors of node i;
$succ(i)$	: $=\{j \mid (i,j)\in A\}$, successors of node i;
D	: destination node that receive the evacuees who initially stay in node s;
T	: required total evacuation time;
$G_T=(N_T,A_T)$	: the time expansion of $G(N,A)$ over a time horizon T;
N_T	: $=\{i(t) \mid i\in N; t=0,1,\ldots,T\}$;
A_T	: set of arcs over a time horizon T;
$f_{ij}(t)$	: flow (number of evacuees) that leave node i at time t and reach node j at time $t+\lambda_{ij}$ without rebalancing solution, $\forall i,j\in N, k=1,\ldots,Q$;
$y_i(t+1)$	: $=f_{i(t),i(t+1)}, \forall i\in N, t=0,\ldots,T$, represents the number of evacuees who prefer to stay in node i at time t for at least one unit time without rebalancing solution;
$f_{ij}^{'}(t)$	: flow (number of evacuees) that leave node i at time t and reach node j at time $t+\lambda_{ij}$ with rebalancing solution, $\forall i,j\in N, k=1,\ldots,Q$;
$y_i^{'}(t+1)$	: $=f_{i(t),i(t+1)}, \forall i\in N, t=0,\ldots,T$, represents the number of evacuees who prefer to stay in node i at time t for at least one unit time without rebalancing solution;
x_{ij}	: the state of flow from i to j on the arc (i,j);
(x_{ij},x_{ji})	: flow pattern between the node i and j;
F	: $=\{(x_{ij},x_{ji}) \mid \forall (i,j)\in A\}$, flow pattern of the network, which is the decision variable of the model.

6.4.3.2 Objective function and constraints

As expressed in the model in Section 6.3.1, the largest imbalanced situation that can be rebalanced depends on two evacuation abilities: (1) the evacuation ability with the original transportation system, which is called evacuation ability 1, and (2) the evacuation ability of the transportation system after an optimal flow pattern planning, which is called evacuation ability 2. Given a required total evacuation time, the evacuation ability 1 will be a constant, which is the result of the optimization of Eq. (6.13);

$$\max \sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}(t) \tag{6.13}$$

The evacuation ability 2 depends on the rebalancing solution of an optimal flow pattern, which is expressed by

$$\max \sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}^{'}(t) \tag{6.14}$$

The objective function and constraints are given below;

$$\text{Max} \quad \frac{\sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}^{'}(t) - \max \sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}(t)}{\sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}^{'}(t)} \tag{6.15}$$

$$\text{s.t.} \quad \sum_{t=0}^{T} \sum_{i \in succ(s)} f_{si}(t) = \sum_{t=0}^{T} \sum_{i \in pred(d)} f_{id}(t) \tag{6.16}$$

$$y_i(t+1) - y_i(t) = \sum_{l \in pred(i)} f_{li}(t-\lambda_{li}) - \sum_{j \in succ(i)} f_{ij}(t), \ \forall i \in N, t = 0, \ldots, T \tag{6.17}$$

$$y_i(0) = 0, \ \forall i \in N, i \neq s \tag{6.18}$$

$$0 \leq y_i(t) \leq a_i, t = 1, \ldots, T, \forall i \in N \tag{6.19}$$

$$x_{ij} = 0,1 \qquad \forall (i,j) \in A \tag{6.20}$$

$$\bar{x}_{ij} = \begin{cases} 0, & x_{ij} = 1 \\ 1, & x_{ij} = 0 \end{cases}, \forall (i,j) \in A \tag{6.21}$$

$$\sum_{t=0}^{T}\sum_{i\in succ(s)} f_{si}^{'}(t)=\sum_{t=0}^{T}\sum_{i\in pred(d)} f_{id}^{'}(t) \tag{6.22}$$

$$y_i^{'}(t+1)-y_i^{'}(t)=\sum_{l\in pred(i)} f_{li}^{'}(t-\lambda_{li})-\sum_{j\in succ(i)} f_{ij}^{'}(t),\ \forall i\in N, t=0,\ldots,T \tag{6.23}$$

$$y_i^{'}(0)=0,\ \forall i\in N, i\neq s \tag{6.24}$$

$$0\leq y_i^{'}(t)\leq a_i, t=1,\ldots,T, \forall i\in N \tag{6.25}$$

$$0\leq f_{ij}^{'}(t)\leq x_{ij}c_{ij}+\bar{x}_{ji}c_{ji}, \forall(i,j)\in A \tag{6.26}$$

$$x_{si}=1, x_{is}=0, \forall(s,i)\in A \tag{6.27}$$

$$x_{id}=1, x_{di}=0, \forall(i,d)\in A \tag{6.28}$$

In the above, Eq. (6.15) represents the resilience of the system, namely maximization of the imbalanced situations. Eq. (6.16) to Eq. (6.19) represent the constraints of evacuation with the original transportation system. Eq. (6.16) represents the constraint of evacuation with the original transportation system that the number of victims evacuated from the source node equals to the number of all victims evacuated to the destination node; Eq. (6.17) calculates the number of victims who stay in node i at time t+1; Eq. (6.18) represents the constraint that there are no victims in the nodes at time 0 except the source node; Eq. (6.19) represents the node capacity.

Eq. (6.20) to Eq. (6.28) represent the constraints of evacuation with flow pattern planning. Eq. (6.20) represents the constraint that x_{ij} is a 0-1 variable; Eq. (6.21) represents the constraint that $\bar{x}_{ij}$ is the opposite of x_{ij} ; Eq. (6.22) represents the constraint that the number of victims that are evacuated from the source node equals to the number of the victims are all evacuated to the destination node; Eq. (6.23) calculates the number of victims who stay in node i at time t+1; Eq. (6.24) represents the constraint that there are no victims in the nodes at time 0 except the source node; Eq. (6.25) represents the node capacity; Eq. (6.26) represents the arc capacity; Eq. (6.27) represents the flow pattern of the roads associated with the source node *s*; and Eq. (6.28) represents the flow pattern of the roads associated with the sink node *d*.

6.4.4 Solving the Model

6.4.4.1 Sketch of the proposed method

The objective function (6.15) is the combination of two separate equations, namely Eq. (6.13) and Eq. (6.14).

Eq. (6.13) describes the evacuation ability of the original transportation system. For a given transportation system and a given demand (i.e., total evacuation time), Eq. (6.13) with its constraints (6.16)-(6.19) is viewed as a minimum cost flow problem, which has nothing to do with the rebalancing solution of flow pattern planning. Thus, Eq. (6.13) can be obtained by an existing minimum cost flow algorithm, e.g. dual ascent algorithm or RelaxIV [Bertsekas and P. Tseng 1994] and is treated as a constant in the objective function (6.15). In other words, the maximal imbalanced situation that can be rebalanced only depends on the optimal rebalancing solution. Details of the optimization of Eq. (6.13) are shown in Fig. 6.9, which is called MNE (Maximum Number of Evacuees) algorithm. In Fig. 6.9, MCFA, *TET* and MNE represent minimum cost flow algorithm, total evacuation time and maximum number of evacuees who can be evacuated within the required evacuation time *T*, respectively.

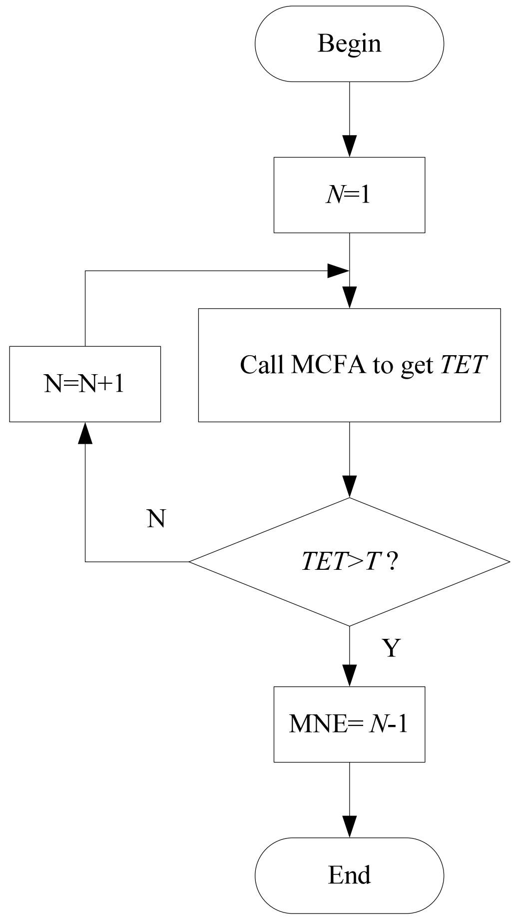

Figure 6.9 MNE algorithm

With the result of Eq. (6.13), the objective function (6.15) (i.e. maximum imbalanced situation) is converted to maximize Eq. (6.14) (i.e., the evacuation ability with an optimal rebalancing solution) with its constraints of (6.20) to (6.28.).

To solve the model, i.e., Eq. (6.15) or Eq. (6.14) with its constraints of (6.20) to (6.28.), a method is proposed to get the optimal rebalancing solution (namely the optimal flow pattern) first, as shown in Fig. 6.10. The proposed method has two layers (upper-layer algorithm and lower-layer algorithm). The two layers are coupled in the sense that in each round of iteration, the upper-layer algorithm will need to call the lower-layer algorithm. The lower-layer algorithm is a minimum cost flow algorithm (MCFA), e.g. dual ascent algorithm or RelaxIV [Bertsekas and P. Tseng 1994]. In this MCFA, we set the number of evacuees as the result of Eq. (6.13). The upper-layer algorithm is to find the optimal flow pattern. It is noted that according to Kim and Shekhar [2005], the planning problem here is a NP-complete problem. Therefore, the upper-layer algorithm for flow pattern planning is an intelligent evolutionary computing algorithm, PSO (particle swarm optimization) in particular. After we find the optimal rebalancing solution, namely the optimal flow pattern, we use the MNE algorithm, as shown in Fig. 6.9, to get the maximum number of evacuees that can be evacuated within given required evacuation time, and further calculate the objective function (6.15).

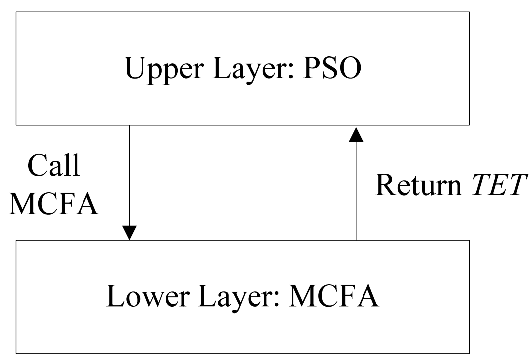

Figure 6.10 Sketch of the proposed method for flow pattern planning

6.4.4.2 PSO algorithm

The PSO algorithm was first developed by Kennedy and Eberhart based on the mimic of a social model [Kennedy and Eberhart 1995, Kennedy and Eberhart 1997]. The algorithm has been widely investigated and applied [Shi and Eberhart 1998, Trelea 2003, Wang and Wang 2008]. The basic PSO algorithm uses a swam of m particles to search a D-

dimensional problem space, and each particle has a velocity ($v_i = (v_{i1}, \cdots, v_{id}, \cdots, v_{iD})$, $1 \le i \le m$, $1 \le d \le D$) and a position ($x_i = (x_{i1}, \cdots, x_{id} \cdots, x_{iD})$) . Details of the basic PSO algorithm are referred to the reference [Kennedy and Eberhart 1995, Wang and Wang 2008]. The basic PSO algorithm is for the continuous variable optimization problem. The PSO algorithm for the discrete variable problem can be found from [Kennedy and Eberhart 1997, Wan et al. 2004]. Our problem is a discrete variable problem, so we apply the discrete PSO algorithm. Details of the PSO for our problem include (1) encoding, (2) update formula, (3) stop criterion, and they are given as follows.

(1) Encoding

Encoding is to design the representation of the position of a particle. Let

n_R : the number of roads in the transportation network;

n_s : the number of roads linked to the source node which is equal to the number of arcs $(s,i), \forall (s,i) \in A$;

n_d : the number of roads linked to the destination node which is equal to the number of arcs $(i,d), \forall (i,d) \in A$.

The number of roads whose flow patterns are unknown is represented as follows:

$$L = n_R - n_s - n_d \qquad (6.29)$$

The position of particle i is represented as:

$$x_i = (x_{i1}, \cdots, x_{ij} \cdots, x_{iL}), 1 \le j \le L \qquad (6.30)$$

where $x_{ij} \in \{0,1,2,3\}$ which will be further explain below.

There is a conversion between the encoding space and the solution space, which is illustrated in Fig. 6.6. In Fig. 6.6, the left part is the code, and the right part is the flow pattern of a road. The mapping function is self-explained in the figure.

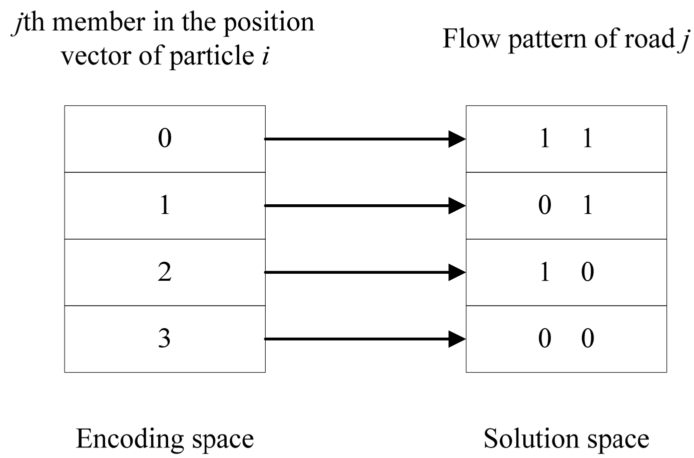

Figure 6.11 Conversion between the encoding space and the solution space

(2) Update formulas

The position vector of a particle is composed of integers. So, the update formulas of the position and the velocity are designed as follows.

$$v_{ij}^{k+1} = \begin{cases} Int(\omega v_{ij}^{k} + c_1 \xi (p_{ij}^{k} - x_{ij}^{k}) + c_2 \eta (p_{gj}^{k} - x_{ij}^{k})), \gamma \le c_0 \\ Int(\omega v_{ij}^{k} + Random(0,4)), r > c_0 \end{cases} \tag{6.31}$$

$$x_{id}^{k+1} = \left| x_{id}^{k} + v_{id}^{k+1} \right| \mathrm{mod}(4) \tag{6.32}$$

In the above, $Int(\cdot)$ is the function to get the integer part of $(\cdot)$; c_1 and c_2 are learning factors, which are two positive constants; $\xi, \eta, \gamma \in U[0,1]$, which are pseudo random numbers with the homogeneous distribution in [0,1], and c_0 is a constant in [0,1], which is used to decide the update mode of the velocity. Eq. (18) means when a random number

γ is not greater than c_0, the velocity is updated according to the traditional mode which is as follows:

$$v_{ij}^{k+1} = Int(\omega v_{ij}^{k} + c_1\xi(p_{ij}^{k} - x_{ij}^{k}) + c_2\eta(p_{gj}^{k} - x_{ij}^{k})) \tag{6.33}$$

When γ is greater than c_0, the velocity is updated according to the random mode which is as follows:

$$v_{ij}^{k+1} = Int(\omega v_{ij}^{k} + Random(0,4)) \tag{6.33}$$

where $Random(0,4)$ is a random generator between 0 and 4. Further, the mod number is used to prevent the particle from flying out of the boundary. The absolute value of $|\cdot|$ is used to keep the position non-negative.

(3) Stop criterion

A positive integer *NG* is taken as the maximum iteration number. If the iteration number *k* is greater than *NG*, the algorithm is stopped. *NG* is set based on the trial-and-error for a particular problem.

The entire procedure of the algorithm has the following steps.

Step 1: Set c_1, c_2 and c_0. Initialize a swarm including *m* particles with random positions inside the solution space. x^* is the best solution and $f(x^*)$ is the best fitness value.

Step 2: Call MCFA to evaluate the fitness of each particle. That the index of the one gets the best fitness is set as *g*. The position and fitness of the particle *g* are used to update x^* and $f(x^*)$.

Step 3: Update the velocity and position of particles according to Eqs (6.31) and (6.32).

Step 4: If the stop criterion is satisfied the algorithm ends; otherwise go to Step 2.

6.4.5 Experiment

6.4.5.1 Experimental setting

Fig. 6.12 illustrates a transportation system which has been partially damaged by losing the lane 4 ->5. There are seven places. The transportation system needs to evacuate victims from Place 1, which is the source node, to Place 7, which is the sink node. The capacity of a place is represented by a bracket "{}" in Fig. 6.12. For example, the capacity of Place 1 is 100; the capacity of other six places can be seen, respectively, from Fig. 6.12. The "-" in the bracket adjacent to Place 7 means that Place 7 is a shelter and its capacity is regarded as infinitely large. The road and lane are expressed by connection between places (see Fig. 6.12). Both capacity and travel time of a lane are indicted on the arc by a parenthesis "( )". For example, Lane 1->2 has travel time 2 and capacity 3, and Lane 2->1 has travel time 2 and capacity 4. Place 1 has N evacuees, which means that there are N people who need to be evacuated. The evacuation demand is such that all the victims must be transferred from Place 1 to Place 7 within 15 time units. It is noted that the number 15 here is the total evacuation time (*TET*) – see the previous definition in Section 6.4.3.

The algorithm has the following settings: the swarm size m=5; the maximum iteration number NG=10. The algorithm is programmed by java and the experiment is performed on a computer with a dual-1.66GHz CPU and a 1.5GB memory. Further, the algorithm runs 10 times for the settings.

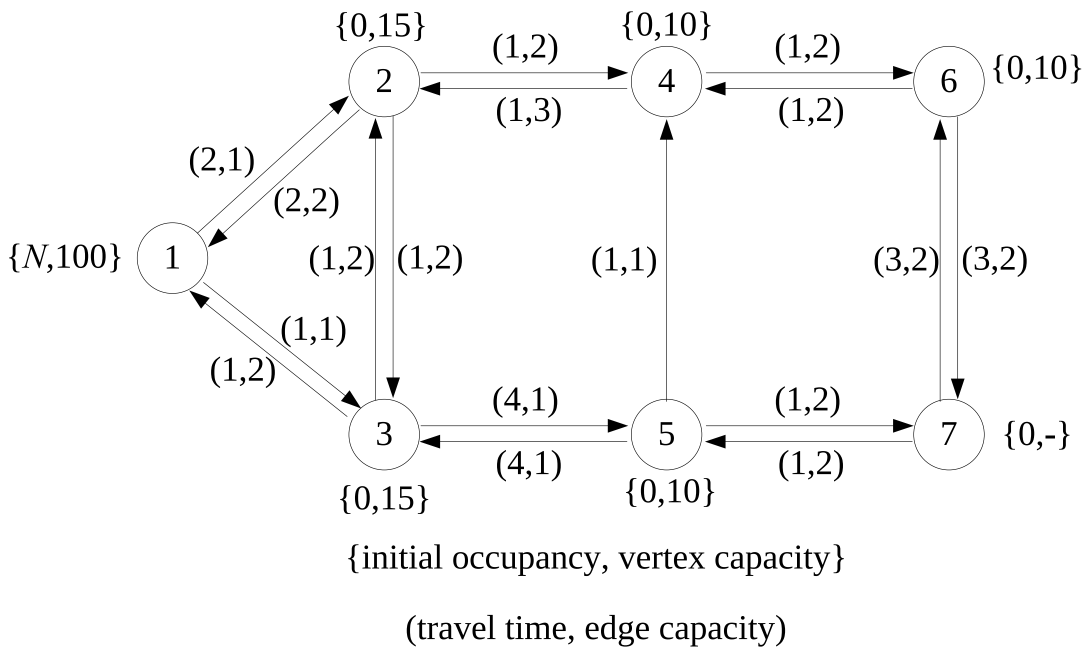

Figure 6.12 Damaged transportation system (the source node is 1; the destination node is 7)

6.4.5.2 Results and discussion

(1) Results

First, the MNE algorithm is used to optimize Eq. (6.13) to find the maximum number of evacuees who can be evacuated by the original damaged transportation system. The result is 19. Therefore, the maximum evacuation ability of the original transportation system is evacuating 19 evacuees within 15 time units.

Second, the two-layer algorithm is used to find the maximum number of evacuees who can be evacuated by the transportation system with the rebalancing solution of flow pattern planning. The result is 38. Therefore, the evacuation ability of the transportation system with the optimal rebalancing solution is evacuating 38 evacuees within 15 time units. Input the two numbers of evacuation ability, namely 19 and 38, into the objective function and we obtain that the maximum imbalanced situation of the system that can be rebalanced is (38-19)/38=50%, which is used to express the resilience of the original transportation system. It is noted that the optimal rebalancing solution for resilience is not unique; the two optimal rebalancing solutions of flow patterns are shown in Fig. 6. 13.

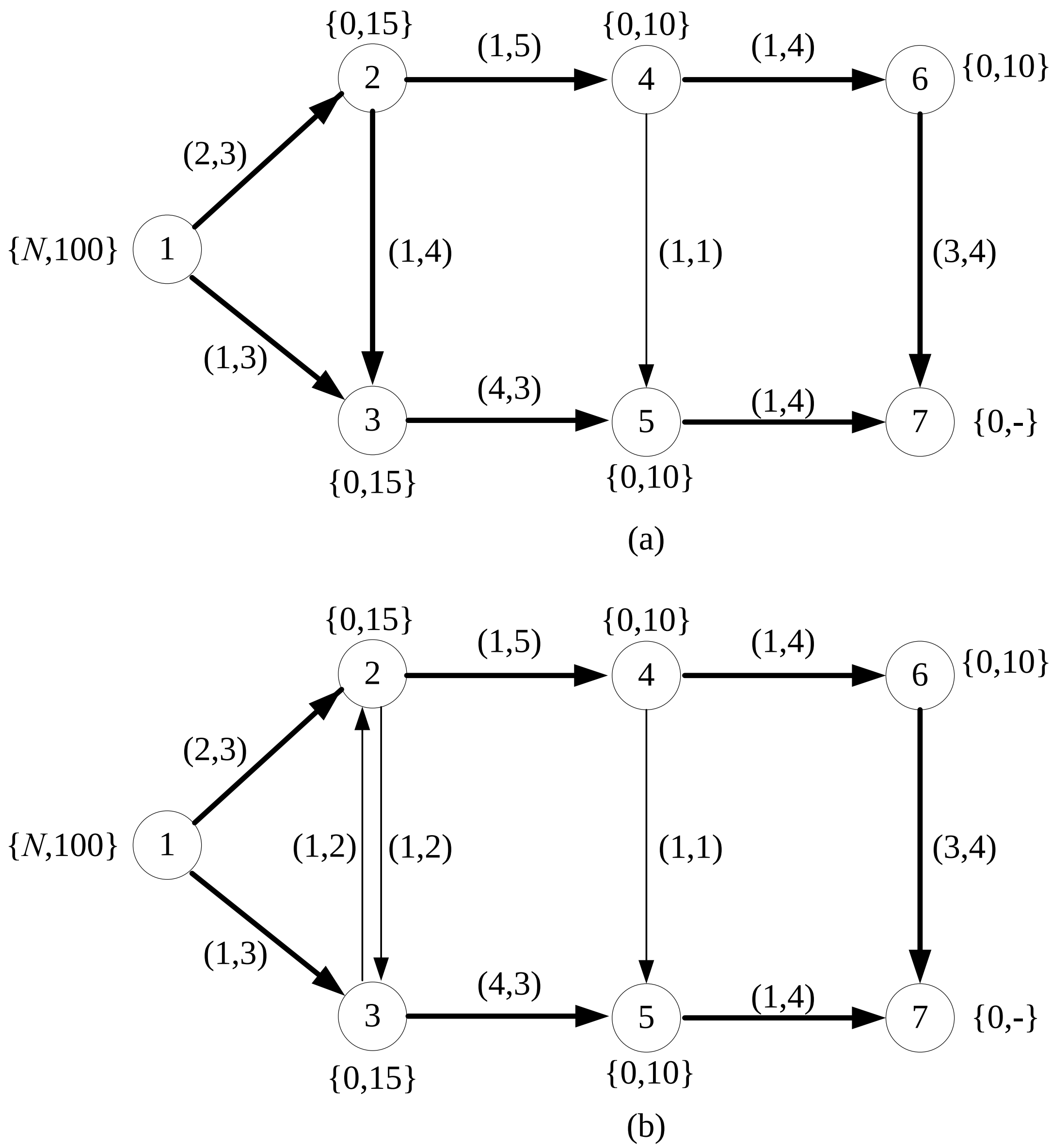

Figure 6.13 Two optimal flow patterns for the system

(2) Discussion

The results of this example indicate that 50% difference between the evacuation demand and supply compared to the evacuation demand could be rebalanced with the rebalancing solution of flow pattern planning. Thus, the resiliency of the example system gives a good explanation of axiom 1 for resilience measurement, as shown in Fig. 6.14.

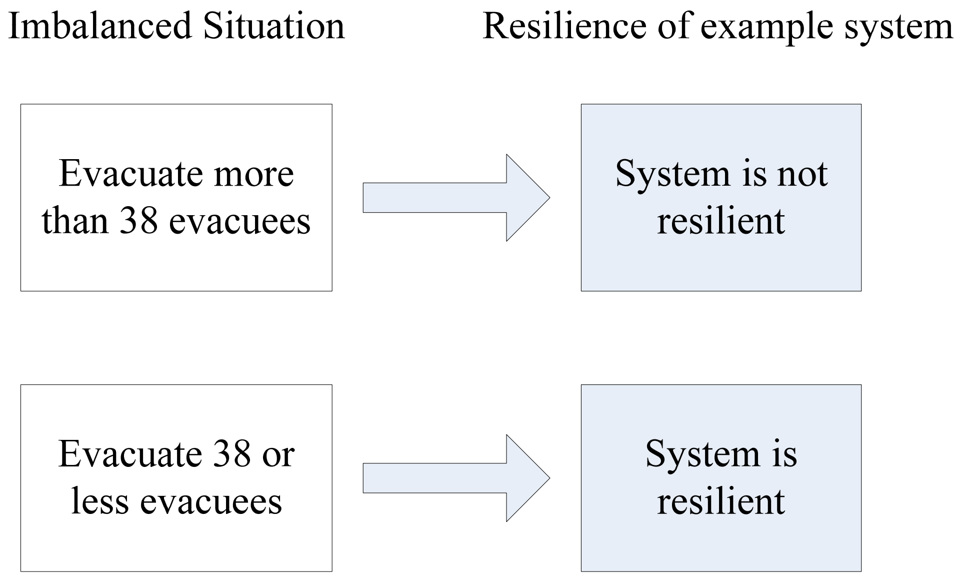

Figure 6.14 Resilience of the example system

Next, illustrations are given for the size of the search space for the given network. The example network has seven nodes. In the lower-layer algorithm, the original network is converted to a time expanded network (TEN) whose size is related to T (required evacuation time). T is 15 in the experiment. As such, the TEN has 106 nodes. In the upper-layer algorithm, one solution represents a flow pattern of five roads, and the whole search space has 1024 solutions.

6.4.6 Conclusion

This section uses an example of transportation system to validate the approach of resilience measurement. A mathematical model based on axiom 2 for resilience measurement was formulated to maximize the imbalanced situation that can be rebalanced with rebalancing solutions. The rebalancing solution only considered flow pattern planning, which is one of the most important methods to increase the evacuation ability of a transportation system in emergency both in theory [Wolshon et al. 2005] and in practice [Dong and Xue 1997, Xue and Dong 2000, Kim and Shekhar 2005, Kim et al. 2008].

There are some limitations in the rebalancing solution expressed in the model. First, the contraflow method in the real-world application is rooted in a single lane; yet our model only captures a group of lanes. Therefore, the model cannot describe the situation where lanes are different in one group which has one flow direction. Second, in the implementation of the contraflow method in practice, there is a setup time (i.e., the time to reverse the direction of flow). This time has not been considered in the presented model. Third, the scalability issue is to be addressed, though our experience is that the computation overhead seems to be manageable.

6.5 Validation Case 2: Enterprise Information System

This section applies the proposed methodology to measure the resilience of an enterprise information system for an imbalanced situation.

6.5.1 Enterprise Information System

As a typical service system, an enterprise information system has two subsystems: infrastructure system (IS) and substance system (SS). The infrastructure system includes the computers, network devices, databases and so on, and the substance system refers to the information which "flows" over the infrastructure. The dynamics of an enterprise information system is determined by the flow of the SS under the constraints of the IS.

An enterprise information system is usually a widely distributed operational information system. As discussed in Section 6.3.2, an enterprise information system is a very special service system in that such a system usually has backup and could be rebalanced from even 100% loss of functions. Providing high availability for such a system is complex for several reasons according to [Cai et al. 2006]: (1) failures at any components may cause cascading failures of the entire system due to distributed information flows [Zhang 2008]; (2) distributed resources may be shared by multiple flows; (3) insufficient resources for system safety due to intensive processing requirement; (4) high requirement of recovery time for information flows. It is noted that the failures mentioned here, including transient failures and non-transient failures, refer to the software failure in the context of the enterprise information system [Gavrilovska et al. 2002]. The recovery from these failures is only related to the software-based recovery without consideration of physical damage or the IS damage. This section extends the software-based recovery to the physical damage. The physical damage cannot usually be recovered with the software-based approach; on the contrary, it needs extra resources, including computers, network devices, and human resources to recover.

6.5.2 Problem Definition

6.5.2.1 Imbalanced situation and rebalancing solutions

We consider a scenario that an enterprise information system is fully damaged. There are m functions in the system and n categories of resources to recover the functions. The number of resources j ($j = 1, 2, \cdots, n$) needed for the recovery of function i ($i = 1, 2, \cdots, m$) is denoted by r_{ij}. The amount of resources for each category is limited. The process time needed for the recovery of function i is p_i. We assume that functions are not recovered at the same time. This assumption implies that the recovery may take a particular order among a set of recovery tasks and the rebalancing solution of the whole enterprise information system depends on this recovery order. For function i, there is a completion time, denoted by c_i. It is noted that the completion time c_i is different from the process time p_i in that the completion is the sum of the starting time for function i and its process time**.** The order of the recovery of different functions are denoted by *S*, which is represented by

$$S = \left[(1), \cdots, (i), \cdots, (m)\right] \qquad (6.34)$$

where (i) refers to the *ith* function to be recovered. Obviously, the completion time c_i is the function of *S*. Therefore, c_i is further represented by $c_i(S)$. The required time for the recovery of function i is represented by d_i.

We further consider that the enterprise information system is resilient, which means the system can be recovered from the imbalanced situation within the required time.

6.5.2.2 Measurement of resilience

The enterprise information system need to be rebalanced within the required time from fully damaged situation with an order of recovery of different functions. Thus, axiom 3 and its corresponding mathematical model should be applied; in particular, the resilience is described with a minimization of the rebalancing time.

6.5.3 Mathematical Model

6.5.3.1 Variable definition

The variables are defined in Table 6.7.

Table 6.7 Variable definition in measurement model for enterprise information system

m	: the total number of functions;
n	: the number of categories of resources;
w_i	: weight of function i, $i=1,2,\cdots,m$, $\sum_{i=1}^{m} w_i = 1$;
r_{ij}	: the number of resource j needed by the recovery of function i, $i=1,2,\cdots,m$, $j=1,2,\cdots,n$;
R_j	: the total number of resource j, $j=1,2,\cdots,n$;
P_i	: the process time for the recovery of function i, $i=1,2,\cdots,m$;
S	: $\{(1),(2),\cdots,(i),\cdots,(m)\}$, sequence of recovery;
$C_i(S)$	: the completion time for the recovery of function i, $i=1,2,\cdots,m$;
T_i	: the required time for the recovery of function i, $i=1,2,\cdots,m$;

6.5.3.2 Objective function and constraints

$$\text{Min} \quad \sum_{i=1}^{m} \frac{C_{(i)}(S)}{w_i T_i} \tag{6.35}$$

$$\text{s.t.} \quad C_{(1)} = P_{(1)} \tag{6.36}$$

$$C_{(i+k)} = \left\{ C_{(i)} + P_{(i+k)} \middle| \sum_{l=0}^{k} r_{(i+l),j} \le R_j,\ j = 1, 2, \cdots, n \right\}, \tag{6.37}$$
$$i = 1, 2, \cdots, m, \quad 1 \le k \le m - i$$

$$C_{(i)}(S) \le T_i, i = 1, 2, \cdots, m \tag{6.38}$$

In the above, Eq. (6.35) represents the minimization of the rebalancing time; Eq. (6.36) represents the completion time of the first function; Eq. (6.37) represents the resource constraints; Eq. (6.38) represents the constraint of recovery time.

6.5.4 Solving the Model

Since the model includes non-analytical constraints and the objective function cannot be computed analytically, the model may be solved by evolutionary algorithms. The genetic algorithm (GA) can be employed for solving the model. The entire algorithm has two layers of solvers, as shown in Fig. 6.15. The first layer contains GA which finds the best task recovery schedule or sequence and the second layer contains RS (Recovery Schedule) which computes the recovery schedule and completion time for each function.

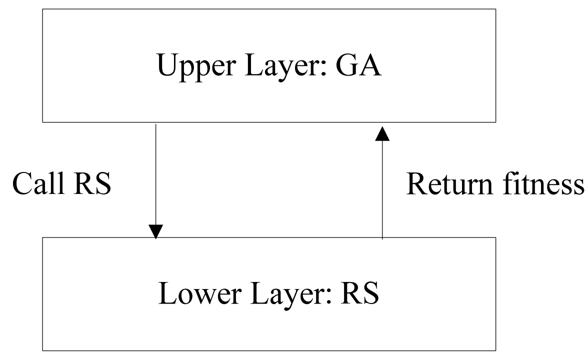

Figure 6.15 Sketch of the two-layer algorithm

There are two key issues in the implementation of the GA for this model, which are: (1) coding and (2) handling of the constraint (6.37). For the first issue, coding scheme is designed as $\left[(1),(2),\cdots,(i),\cdots,(m)\right]$, where (i) represents a number between 1 and m. For the second issue, the solutions that do not satisfy constraint (6.37) will simply be abandoned in the iteration.

The RS algorithm is designed as follows (given a sequence $\vec{v}=\{(0),(1),\cdots,(m)\}$):

Step 1: Let $c_{(0)}=0$, $\vec{v}=\{(0),(1),\cdots,(m)\}$, $CT_j=R_j-r_{(1),j}, j=1,2,\cdots,n$, $i=1$, $p=0$, $c_{(i)}=p_{(i)}$.

Step 2: $i=i+1$, if $i>m$, stop; else, go to Step 3.

Step 3: If $r_{(i),j}\le CT_j, \forall 1\le j\le n$, go to Step 4; else go to Step 7.

Step 4: $c_{(i)}=c_{v_p}+p_{(i)}$, $CT_j=CT_j-r_{(i),j}, j=1,2,\cdots,n$, $k=i$.

Step 5: If $c_{v_k}<c_{v_{k-1}}$, go to Step 6.

Step 6: $x = v_k$, $v_k = v_{k-1}$, $v_{k-1} = x$, $k = k - 1$. If $k = p$, go to Step 2; else go to Step 5.

Step 7: $p = p + 1$, $CT_j = CT_j + r_{v_p, j}$, $j = 1, 2, \cdots, n$, go to Step 3.

In the RS algorithm, the definitions of $c_{(i)}$, $p_{(i)}$, $r_{(i),j}$, R_j are the same as those in Section 3.1. CT_j, *p*, *k*, *x* are temporary variables in the algorithm.

6.5.5 Experiment

To illustrate how the model and algorithm work, an example is employed. Table 6.8 gives the information about function recovery for a particular enterprise information system. There are five functions in this system, which have been damaged by the outer attack. There are two categories of resources. The total number of resource 1 is 7; the total number of resource 2 is 10. All the functions are treated with the same importance and the weight for each function is 0.2 (equal weighting).

Table 6.8 Recovery parameters

Function	Resource 1 $r_{i,1}$	Resource 2 $r_{i,2}$	P_i (min)	T_i (min)
i=1	2	3	2	7
i=2	2	4	5	8
i=3	4	3	3	8
i=4	2	3	3	10
i=5	3	4	4	10

The original solution space has 120 solutions. The optimum solutions for this problem are:[1,2,4,5,3], [1,4,2,5,3], [2,1,4,5,3], [2,4,1,5,3], [4,1,2,5,3], [4,2,1,5,3], respectively.

All these solutions result in the same optimum recovery schedule and the same fitness value. According to the lower-layer algorithm RS, we can get the optimum recovery schedule, which is: (1) function 1, function 2 and function 4 are recovered first at the same time and will be completed after 2 minutes, 5 minutes and 3 minutes, respectively, (2) then, function 5 will be recovered immediately after the recovery of function 1 is completed and will be completed after another 4 minutes which indicates the total recovery time of function 5 is 6 minutes, and (3) last, function 3 will be recovered after the recovery of function 5 is completed and the total recovery time of function 3 is 8 minutes. The fitness value of this optimal schedule is 0.5621. This result implies that the functions of the enterprise information system can be recovered within 56.21% of the required time.

6.5.6 Conclusion

This section uses an example of enterprise information system to validate the approach of resilience measurement. A mathematical model based on axiom 3 of the resilience measurement is formulated to minimize the rebalancing time. It is remarked that the proposed measure for resilience is similar to the production scheduling problem. However, the result of the example indicates that the optimum recovery schedule or sequence does not follow the shortest processing time (SPT) rule or earliest due-date (EDD) rule in the production scheduling problem [Baker and Trietsch 2009]. In fact, this does not come to a surprise, as there are two significant differences with these two problems. First, the SPT and EDD rules are only applicable to a single source scheduling problem; whereas the resilience measurement problem is essentially a multiple resource scheduling problem. Second, the SPT or EDD schedule is optimal if the objective is to minimize the average flow time; whereas the resilience measurement problem not only considers the completion time for function recovery but also the required time.

6.6 Conclusions

This chapter proposed the axioms of the resilience measurement based on the definition of resilience in Chapter 5. Different mathematical models based on different axioms were presented. Two example systems, transportation system and enterprise information system, were used to validate the proposed methodology.

In Section 6.4, the rebalancing solution of a transportation system only considered planning the substance of the system, e.g. flow pattern planning. In Section 6.5, the rebalancing solution of an enterprise information system only considered the resource reallocation for different recovery solutions. However, these rebalancing solutions may not always meet the resilience requirement of a service system, and other rebalancing solutions are necessary. For example, in an enterprise information system, the infrastructure reconfiguration is also an important means to recover the damaged functions. The next chapter will address this issue in the context of integrated design and operation of service systems.

CHPATER 7
IMPROVEMENT OF RESILIENCE

7.1 Introduction

As discussed in Chapter 5, improvement of the resilience performance of a service system simply means to generate a better rebalancing solution; in particular, from an engineering perspective, the rebalancing solutions focus on reconfiguration of resources. This chapter presents a methodology for improving the resilience property of a service system by (1) proposing a conceptual model as well as a mathematical model for reconfiguration of resources by integration of design, planning and management, and (2) developing algorithms to solve the mathematical model. Therefore, in Section 7.2, a conceptual model for integrated design, planning, and management of a service system for resilience is presented along with a general computational model. In Section 7.3, a specific model with its algorithms is presented. At last, conclusions are given in Section 7.4.

7.2 Integration Strategy of Resource Reconfiguration

This section proposes a system level methodology of resource reconfiguration by integration of design, planning and management.

7.2.1 Design, Planning and Management

First, three reconfiguration strategies, design, planning and management are defined.

The **design** of a service system is defined as reconfiguration of infrastructure of the service system; in particular, it is meant to determine the parameters of the infrastructure, including the parameters of the nodes and arcs. Fig. 7.1 is an example of design as the rebalancing solution for a damaged wireless communication system in an imbalanced situation [sinonet 2012]. In Fig. 7.1, a wireless communication system was damaged due to a flood happened in Beijing in 2012 and it can not meet the demand of customers. Thus, one new node was built into the communication network by a temporary emergency wireless communication vehicle.

Figure 7.1 Example of design in rebalancing solution
(adopted from http://www.sinonet.org/news/china/2012-07-27/216300_2.html)

The **planning** of a service system is defined as reconfiguration of substance of the service system; in particular, it is meant to determine the parameters of the substance flows and how the substance flows over the infrastructure. Fig. 7.2 is an example of flow pattern planning with contraflow method as the rebalancing solution for a transportation system in an imbalanced situation in Houston during Hurricane Rita evacuation [Kim et al. 2007]. In Fig. 7.2, (a) is the network before contraflow and (b) is the network after contraflow.

(a) Before contraflow (b) After contraflow

Figure 7.2 Example of planning in rebalancing solution [Kim et al. 2007]

The **management** of a service system is defined as reconfiguration of extra input resources which are used to reconfigure of infrastructure and substance; in particular, it is meant to determine (1) the parameters of extra resources input into the system, say task 1, and (2) the reconfiguration of these resources in IS and SS, say task 2. Task 1 is performed according to the difference between supply and demand. If a supply function does not meet the demand function, the corresponding resources to improve this supply function will be input into the system. Thus, task 1 determines the types of resources and the amounts of each type. Task 2 determines how to use the input resources, namely configuration of these resources in IS and SS. Obviously, management is coupled with the other two reconfigurations, namely design and planning. Fig. 7.3 is an example of a school bus (input resource) was assigned to evacuate victims from forest fires in northern Saskatchewan in 2008 [CTV 2008].

Figure 7.3 Example of evacuation management in rebalancing solution [CTV 2008, http://www.ctv.ca/servlet/ArticleNews/story/CTVNews/20080703/fire_sask_080703/20080703/]

It is noted that the three definitions above use the term "reconfiguration", not configuration. The reason is that in a particular imbalanced situation, a service system has already been in a status that all the resources are configured and a rebalancing solution is generated by reconfiguring some resources. Fig. 7.4 summarizes an illustration of three reconfigurations.

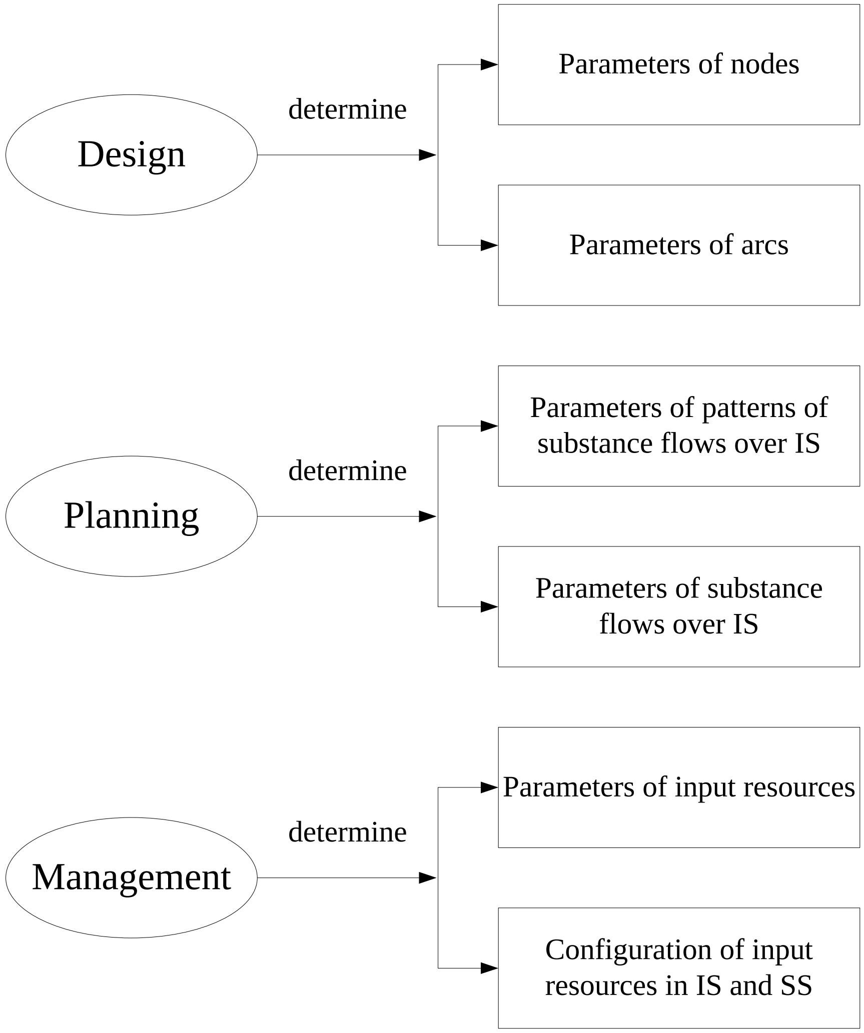

Figure 7.4 Design, planning and management

7.2.2 Integration Framework

A reconfiguration strategy on different resources is expressed into an integration framework, as shown in Figure 7.5.

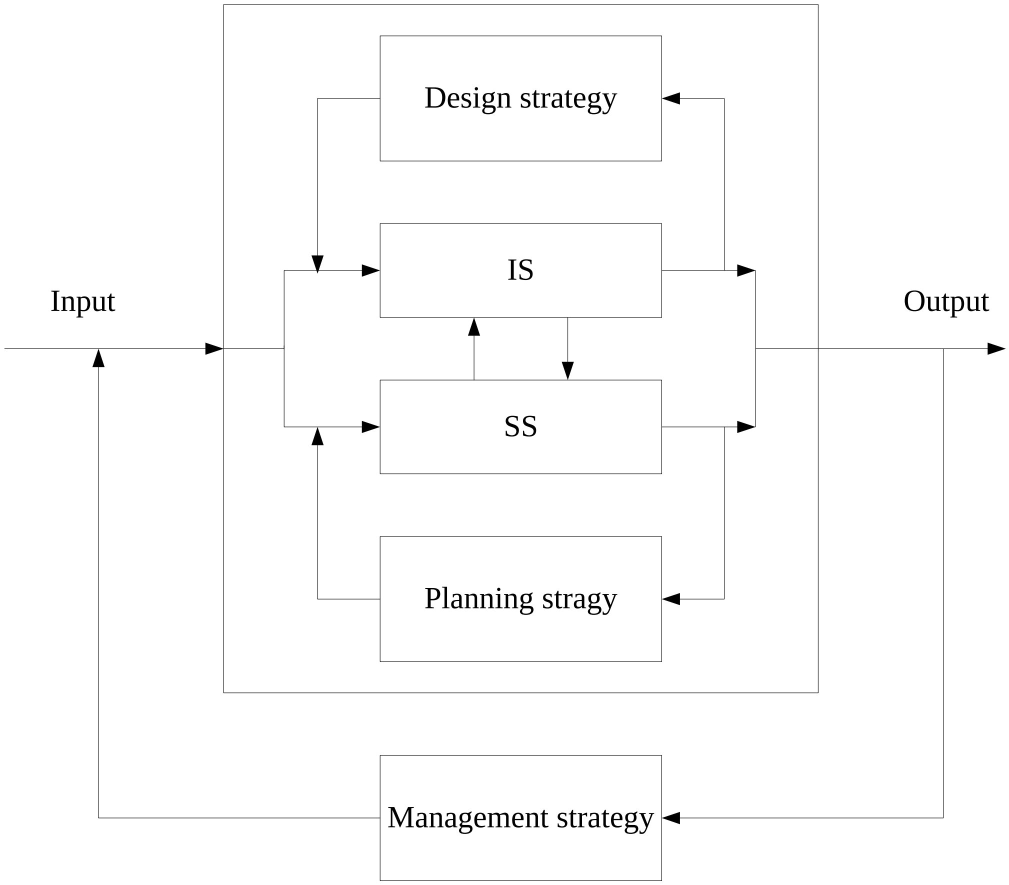

Figure 7.5 A framework of integration of design, planning and management

In Figure 7.5, IS refers to the infrastructure system and SS refers to the substance system. The Input includes extra input resources and input information (demand functions). The out refers to the supply functions. Next, design, planning and management are given detailed definitions for a service system in the context of rebalancing solutions.

Figure 7.5 implies that the rebalance of a service system to make the supply functions meet the demand functions has two options: (1) to reconfigure the resources within a service system, namely IS and SS, to improve the supplied functions, and (2) to reconfigure both resources within the service system and extra resources from the input to improve the supply functions.

Several remarks are given below for further explanation on the framework.

Remark 1: Three types of resources are considered in the resilience analysis: IS, SS and input resources.

Remark 2: Three types of reconfigurations are considered to increase the supply functions: design, planning and management.

Remark 3: Three types of reconfigurations show the potential ability of resilience. The three types of reconfigurations may be performed separately or be performed simultaneously. It is noted that in the current literature, the three reconfigurations are usually examined separately. However, it is obvious that integration of them as a whole could improve resilience.

Remark 4: The regular reconfiguration strategies, design, planning and management that work in a separate manner, could be viewed as the special cases of the integrated manner. Another three possible special cases of the integration framework are: integration of design and planning, integration of design and management, integration of planning and management.

7.2.3 Mathematical Model of Integration Strategy

This section presents a general mathematical model to formulate the integration strategy of resource reconfiguration.

7.2.3.1 Assumptions

The model considers the integration of design, planning and management. As the goal of this model is to find the maximum ability of resilience, all the available input resources are used in the management strategy. Thus, task 1 of management (see definition in Section 7.2.1.1) is not considered. The assumptions of the model are given below.

(1) Given a networked service system in an imbalanced situation.

(2) Given required rebalancing time.

(3) Three reconfiguration activities, namely design, planning and management approaches are considered in the rebalancing solution.

(4) Multiple types of substance are considered in the substance system.

(5) Given two types of extra input resources for design and planning respectively.

7.2.3.2 Variable definition

The variables of the model are given in Table 7.1 Note that the decision variables are: $\boldsymbol{x_d}$, $\boldsymbol{x_p}$ and $\boldsymbol{x_m}$.

Table 7.1 Variable definition in the model of integration strategy

$G=(N,A)$	: directed network with N set of nodes and A set of arcs;
m	: the total number of functions;
n_d	: the category number of extra input resources for design;
n_p	: the category number of extra input resources for planning;
w_i	: weight of function i, $i=1,2,\cdots,m$, $\sum_{i=1}^{m} w_i = 1$;
$r_{i,j}^{k_1}$	: the number of resource of category k_1 needed by arc (i,j) in design, $(i,j)\in A$, $k_1=1,2,\cdots,n_d$;
$r_j^{k_2}$	: the number of resource of category k_2 needed by node j in planning, $j\in N$, $k_2=1,2,\cdots,n_p$;
$S_i^t(x)$	: the supply function i at time t with rebalancing solution x, $i=1,2,\cdots,m$;
T_i	: the required time for the rebalance of function i, $i=1,2,\cdots,m$;
d_{ij}	: design variable for arc $(i,j)\in A$;
d_j	: design variable for node $j\in N$;
Q	: category number of substance
p_{ij}	: patterns of substance flow over $(i,j)\in A$;
f_{ij}^l	: amount of substance flow of type l over $(i,j)\in A, l=1,\ldots,Q$;
$r_{ij}^{k_1}$	: management variable for arc $(i,j)\in A, k_1=1,\ldots,n_d$;
$r_j^{k_2}$	: management variable for node $j\in N, k_2=1,\ldots,n_p$;
$\boldsymbol{x_{d1}}$	$:=\left\{d_{ij}\,\middle|\,(i,j)\in A\right\}$, design variables for arcs in the network;
$\boldsymbol{x_{d2}}$	$:=\left\{d_j\,\middle|\,j\in N\right\}$, design variables for nodes in the network;
$\boldsymbol{x_{p1}}$	$:=\left\{p_{ij}\,\middle|\,(i,j)\in A\right\}$, planning variables for patterns of substance flows;
$\boldsymbol{x_{p2}}$	$:=\left\{f_{ij}^l\,\middle|\,(i,j)\in A, l=1,\ldots,Q\right\}$, planning variables for amounts of substance flows;
$\boldsymbol{x_{m1}}$	$:=\left\{r_{ij}^{k_1}\,\middle|\,(i,j)\in A, k_1=1,\ldots,n_d\right\}$, management variables for reconfiguration of input resources for design;
$\boldsymbol{x_{m2}}$	$:=\left\{r_j^{k_2}\,\middle|\,j\in N, k_2=1,\ldots,n_p\right\}$, management variables for reconfiguration of input resources for planning;
$\boldsymbol{x_d}$	$:=\boldsymbol{x_{d1}}\cup\boldsymbol{x_{d2}}$, decision variable set for design;
$\boldsymbol{x_p}$	$:=\boldsymbol{x_{p1}}\cup\boldsymbol{x_{p2}}$, decision variable set for planning;
$\boldsymbol{x_m}$	$:=\boldsymbol{x_{m1}}\cup\boldsymbol{x_{m2}}$, decision variable set for management;
x	$:=x\left(\boldsymbol{x_d},\boldsymbol{x_p},\boldsymbol{x_m}\right)$, rebalancing solution

7.2.3.3 Objective function and constraints

The objective is to maximize the supply functions.

Max	$\sum_{i=1}^{m} w_i S_i^{T_i}(x)$	(7.1)
s.t.	$x = x\left(\boldsymbol{x_d}, \boldsymbol{x_p}, \boldsymbol{x_m}\right),\ i = 1, 2, \cdots, m$	(7.2)
	$\boldsymbol{x_d} = \boldsymbol{x_{d1}} \cup \boldsymbol{x_{d2}}$	(7.3)
	$\boldsymbol{x_{d1}} = \left\{ d_{ij} \middle| (i,j) \in A \right\}$	(7.4)
	$\boldsymbol{x_{d2}} = \left\{ d_j \middle| j \in N \right\}$	(7.5)
	$\boldsymbol{x_{p1}} = \left\{ p_{ij} \middle| (i,j) \in A \right\}$	(7.6)
	$\boldsymbol{x_{p2}} = \left\{ f_{ij}^{l} \middle| (i,j) \in A, l = 1, \ldots, Q \right\}$	(7.7)
	$\boldsymbol{x_m} = \boldsymbol{x_{m1}} \cup \boldsymbol{x_{m2}}$	(7.8)
	$\boldsymbol{x_{m1}} = \left\{ r_{ij}^{k_1} \middle| (i,j) \in A, k_1 = 1, \ldots, n_d \right\}$	(7.9)
	$\boldsymbol{x_{m2}} = \left\{ r_{j}^{k_2} \middle| j \in N, k_2 = 1, \ldots, n_p \right\}$	(7.10)
	$\boldsymbol{g_d}\left(\boldsymbol{x_d}, \boldsymbol{x_{m1}}\right) \le 0$	(7.11)
	$\boldsymbol{g_p}\left(\boldsymbol{x_d}, \boldsymbol{x_p}, \boldsymbol{x_{m2}}\right) \le 0$	(7.12)
	$\boldsymbol{g_m}\left(\boldsymbol{x_m}\right) \le 0$	(7.13)

In the above, formula (7.1) represents the supply functions at the corresponding required rebalancing time. Eq. (7.2) represents the rebalancing solution is a function of design variables, planning variables and management variables. Eq. (7.3) represents design variable set consisting of two sets, one for arcs and the other one for nodes, which are further defined in Eq. (7.4) and (7.5), respectively. Eq. (7.6) and (7.7) represent the planning variables for patterns and amounts of substance flow. Eq. (7.8) represents management variable set consisting of two sets, the one for input resources for design and the other one for input resources for planning, which are further defined in Eq. (7.9) and (7.10), respectively. Formulas (7.11), (7.12) and (7.13) represent constraints for design, planning and management, respectively; furthermore, these three formulas imply the coupling relationship among three decision variable sets, namely $\boldsymbol{x}_d, \boldsymbol{x}_p, \boldsymbol{x}_m$.

7.2.3.4 Discussion

The proposed model contains multiple types of variables. The decision variables for design are discrete variables. If only one parameter level is considered, the design variables are binary variables; if multiple parameter levels are considered, the design variables are integer variables. The planning variables are continuous variables. The management variables depend on particular categories of input resources. If the resources are vehicles in a transportation system, the corresponding variable is a integer variable; if the resources are memory space in an enterprise information system, the corresponding variable is a continuous variable.

The proposed model is a multiple layer optimization model. The design is a layer optimization problem, which contains the reconfiguration of IS and configuration of input resources for design. The planning is a two layer of optimization problem, which configures patterns and amounts of substance flows. The management is a layer of optimization problem, which configures resources for design and planning. Therefore, the model generally has four layers of optimization process, which requires heavy computational efforts.

7.3 Validation Case of Transportation System

This section applies the proposed methodology to improve the resilience of a transportation system.

7.3.1 Problem Definition

The scenario is that one place is in a dangerous area (single source) where there are a few categories of victims who are to be evacuated. There are also a few different categories of safe places to which the victims are to be sent. A transportation network lies in between the source and destinations; in particular, the transportation system is partially damaged in an emergency condition. The demand of evacuating the victims is much larger than the regular transportation demand between the source and destinations. Therefore, the transportation system functioning with a regular manner cannot meet the demand, which leads to an imbalanced situation of the system. The general features of the transportation systems are the same as those in Section 6.4.1. Next, the details of the evacuation demands and rebalancing solution are explained.

7.3.1.1 Evacuation demands

The victims on the source place are ranged into different categories based on their characteristics. The destination places are also classified into different categories according to the categories of the victims they receive. We label the different categories of victims with different evacuation priorities. Different categories of victims have different evacuation demands, as shown in Fig. 7.6: (1) the victims of the same category will be evacuated to the corresponding destinations; for instance, the victims who are injured need to be sent to the hospital and the victims without injury need to be sent to normal residence places, and (2) the victims with higher priority will be evacuated earlier; for example, the injured victims need to be evacuated first.

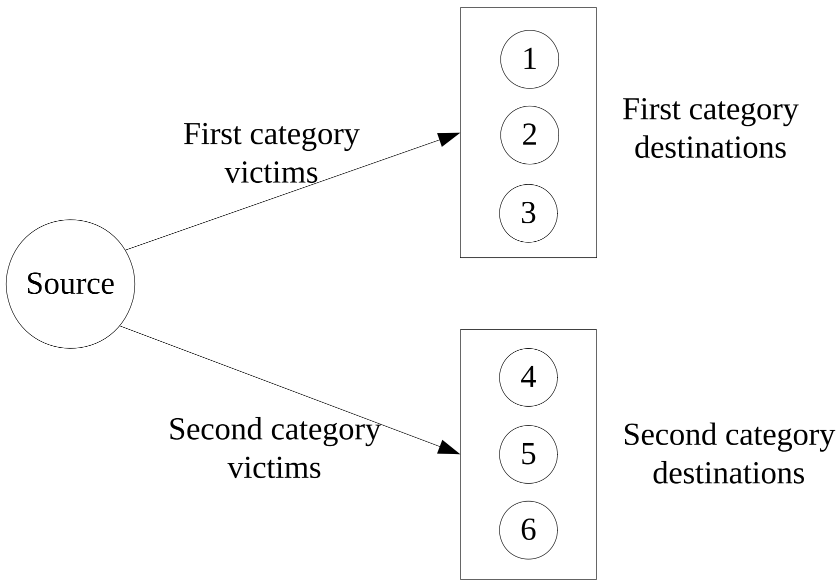

Figure 7.6 Demands of victims of different categories

7.3.1.2 Rebalancing solution with integration of design, planning and management

Integration strategy of design, planning and management is adopted in the transportation system to create an optimal rebalancing solution.

(1) Management. Two categories of input resources are considered in the management strategy: one category for design of new lanes and one category for contraflow implementation. The total amounts of two categories are limited. The configuration of two input resources is coupled with design and planning.

(2) Design. Re-constructing new lanes into the infrastructure is considered in the design strategy. We consider the re-construction of roads as restricted by the following conditions: (1) only limited lanes can be added into a remaining damaged transportation network and each new lane need a fixed amount of resource, (2) each new lane *i->j* falls into the edge domain of the original transportation network, (3) the travel time of the new lane *i->j* is the same as the original travel time t_{ij} of the lane *i->j* in the original transportation network, and (4) the capacity of each new lane is a constant.

(3) Planning. There are two tasks in the planning. One task is to determine the flow pattern of the network. The other task is to determine the amount of flows of different categories over each edges. The definition and expression of flow pattern has been given in Section 6.4. To meet the demand to transport the different categories of victims, we need to plan the flow pattern first for the victims of the first category, and then for the victims of the second category, and so on. If there are Q categories of victims, the planning of the flow pattern will be divided into Q periods. During the *kth* period, the *kth* category has the highest priority to use the transportation network; furthermore, the other categories will be evacuated concurrently when there is extra transportation capacity. In this case, there will be Q flow patterns of the transportation network during the evacuation process, as shown in Fig. 7.7. Usually, the best flow patterns for the victims of different categories are different, so the flow pattern planning is dynamic. The flow pattern for the *kth* category of victims in the *k*th period should not be changed during the time that the victims of the *kth* category are being evacuated. To make use of the transportation network adequately, during the *kth* period of the evacuation process,

besides evacuating the *kth* category victims, the victims whose priorities are less than the *kth* category are also evacuated whenever possible. Furthermore, the setup time and required resource for implementation are considered in contraflow. It is supposed that the resource required to implement contraflow on one road is fixed. Thus, the total number of roads with contraflow is limited due the constraint of input resource.

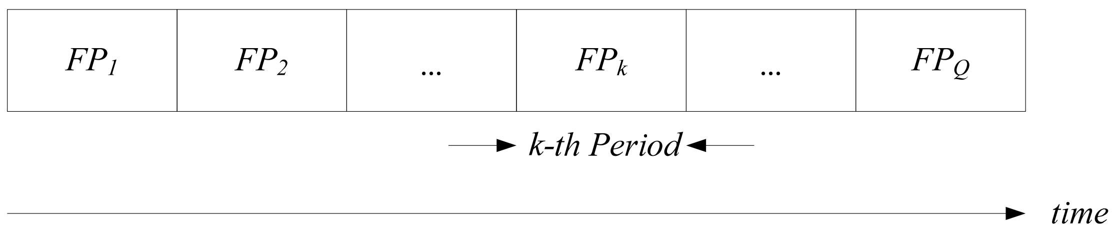

Figure 7.7 Dynamic flow patterns during the whole evacuation process

7.3.1.3 Objectives of the problem

The general goal of the problem is to find the optimal rebalancing solution, namely to minimize the total evacuation time (*TET*). Section 6.4.3 has given definition of total evacuation time. However, the setup time for contraflow has not been considered in the evacuation model in Section 6.4.3. The reason is that it is difficult to estimate the setup time in practice. In this problem, flow patterns change with respect to different categories of victims during the whole evacuation process. This implies that setup time changes from one flow pattern to another. Therefore, this problem has two specific objectives: (1) to minimize the total evacuation time, and (2) to minimize the setup time for flow patterns. It is noted that the two objectives have a conflict, as the first objective essentially requires changing the flow pattern, which thus increases setup time, while the second objective is in favor of no change of the flow pattern. Therefore, this problem is a multiple objective optimization problem.

7.3.2 Mathematical Model

Similar with expression in Section 6.4.3, the flow pattern of road connecting place *i* and place *j* during the *k-th* period can be expressed by $\left(x_{ij}^k, x_{ji}^k\right)$; in particular the four states of the road are represented by $\left(x_{ij}^k, x_{ji}^k\right)$ as shown in Fig. 7.8. The flow pattern of the roads associated with source place and destination places are also predefined as in Section 6.4.3 (Fig. 6.8). It should be noted that when the flow pattern for the victims of the first category is planned, only the destinations of the first category are regarded as the destinations, and the destinations for the victims of the other categories are regarded as the common places.

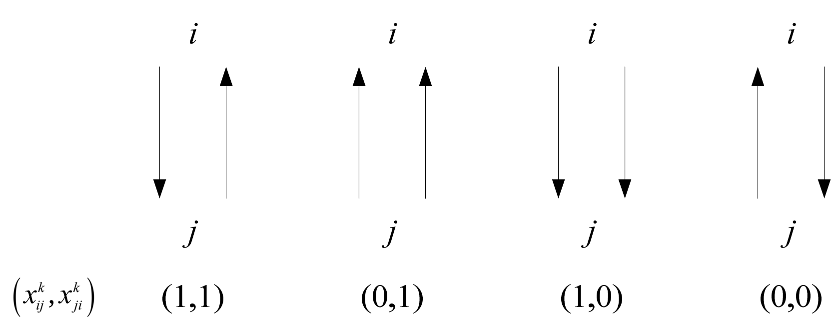

Figure 7.8 Domain of the variable of the flow patterns on a single road

The **individual evacuation time** of one evacuee (ET_i , where i stands for this evacuee) is defined the same as that in Section 6.4.3. The **total evacuation time** of the *kth* category victims (TET_k) is defined as a period of time from the moment when the first evacuee of the first category leaves from a source node to the moment when the last evacuee of the *kth* category arrives at a destination node. Based on these definitions, a horizontal coordinate is added to Fig. 7.7, resulting in Fig. 7.9.

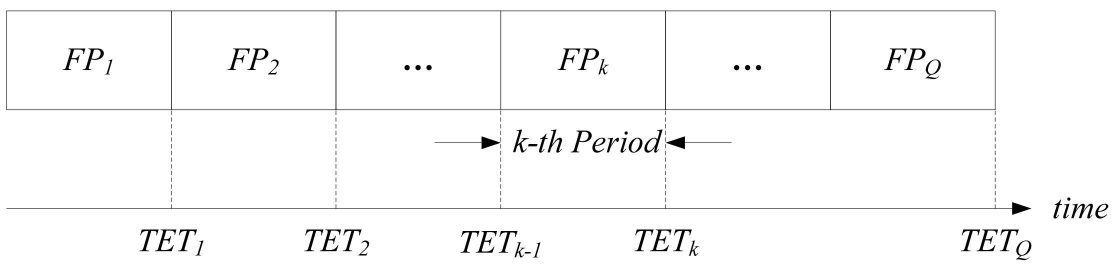

Figure 7.9 Dynamic flow patterns with horizontal coordinate

The first objective function is to minimize the total evacuation time. Fig. 7.9 shows that the total evacuation time of all the victims is the total evacuation of category Q, namely TET_Q.

The second objective function is to minimize the setup time of contraflow. It is noted that setup time is proportional to the number of lanes that need to be reversed. Therefore, minimization of setup time equals minimization of the total number of lanes to be reversed. The second objective function can thus be expressed by

$$\sum_{\forall(i,j)\in A} x_{ij}^{k-1} \oplus x_{ij}^{k}, k = 1, \cdots, Q \qquad (7.14)$$

where $\oplus$ denotes 'XOR' operation; that is, $x_{ij}^{k-1} \oplus x_{ij}^{k}$ means lane x_{ij} needs to be reversed for the *kth* category.

7.3.2.1 Variable definition

The variables are defined in Table 7.2. The decision variables of the model are: $f_{ijk}(t)$, $\left(x_{ij}^{k}, x_{ji}^{k}\right)$ and z_{ij}^{k}.

Table 7.2 Variable definition in resilience improvement model for transportation system

$G=(N,A)$	: directed network with N set of nodes and A set of arcs (static network);
$m=\lvert N\rvert$	: number of elements in set N;
$n=\lvert A\rvert$	: number of elements in set A;
c_{ij}	: capacity of arc (i,j);
λ_{ij}	: travel time of arc $(i,j), \forall (i,j)\in A$;
a_i	: node capacity (the maximum number of evacuees who can stay in node *i*);
s	: source node;
Q	: initial number of evacuees in node *s*;
q_k	: the initial number of evacuees of *kth* category in node *s*, $k=1,\dots,Q$;
$pred(i)$	: $=\{j \mid (j,i)\in A\}$, predecessors of node *i*;
$succ(i)$	: $=\{j \mid (i,j)\in A\}$, successors of node *i*;
D_k	: the set of destinations that receive the evacuees of *k*th category, $k=1,2,\cdots,Q$;
D	: the set of destinations, $D_k \subset D, k=1,2,\cdots,Q$;
T	: required total evacuation time;
$G_T=(N_T,A_T)$	: the time expansion of $G(N,A)$ over a time horizon *T*;
N_T	: $=\{i(t) \mid i\in N; t=0,1,\dots,T\}$;
A_T	: set of arcs over a time horizon *T*;
d	: the super destination node;
$f_{ijk}(t)$	: flow (number of evacuees) of *k*th category that leave node *i* at time *t* and reach node *j* at time $t+\lambda_{ij}, \forall i,j\in N, k=1,\dots,Q$;
$y_i(t+1)$	: $=\sum_{k=1}^{Q} f_{i(t),i(t+1),k}, \forall i\in N, t=0,\dots,T$, represents the number of evacuees who prefer to stay in the node *i* at time *t* for at least one unit time;
x_{ij}^{k}	: the state of flow from *i* to *j* on the arc (*i*,*j*) during the *kth* period,

	$k=1,\ldots,Q$;
$\left(x_{ij}^k, x_{ji}^k\right)$	: flow pattern between the node i and j during the kth period, $k=1,\ldots,Q$;
z_{ij}^k	: add one lane i->j, which capacity is c_n, to arc (i,j) during the kth period, $(i,j)\in A$, $k=1,\ldots,Q$;
r_d	: required amount resource for adding one lane;
R_d	: total amount of input resources for design;
r_p	: required amount of resource for implementation of contraflow in one road;
R_p	: total amount of input resources for flow pattern planning;
c_n	: the capacity of a new lane.

7.3.2.2 Objective functions and constraints

The objective functions and constraints are given below.

$$\min \quad F_1 = TET_Q \tag{7.15}$$

$$\min \quad F_2(k) = \sum_{\forall (i,j)\in A} x_{ij}^{k-1} \oplus x_{ij}^k, k=1,\cdots,Q \tag{7.16}$$

$$\text{s.t.} \quad TET_k = \left\{T \middle| f_{ijk}(t)=0, \forall t>T\right\}, k=1,\ldots,Q;\ TET_0=0 \tag{7.17}$$

$$TET_{k-1} \le TET_k, k=1,\cdots,Q \tag{7.18}$$

$$x_{ij}^k = 0,1 \qquad \forall (i,j)\in A, k=1,\ldots,Q \tag{7.19}$$

$$\bar{x}_{ij}^k = \begin{cases} 0, & x_{ij}^k = 1 \\ 1, & x_{ij}^k = 0 \end{cases}, \forall (i,j)\in A, k=1,\ldots,Q \tag{7.20}$$

$$z_{ij}^k = 0,1 \qquad \forall (i,j)\in A, k=1,\ldots,Q \tag{7.21}$$

$$\bar{z}_{ij}^k = \begin{cases} 0, & z_{ij} = 1 \\ 1, & z_{ij} = 0 \end{cases}, \forall (i,j)\in A, k=1,\ldots,Q \tag{7.22}$$

$$\sum_{\forall (i,j)\in A} r_d z_{ij} = R_d \tag{7.23}$$

$$r_p \sum_{k=1}^{Q} \sum_{\forall (i,j)\in A} x_{ij}^{k-1} \oplus x_{ij}^k \le R_p \tag{7.24}$$

$$\sum_{t=0}^{T} \sum_{i\in succ(s)} f_{sik}(t) = q_k, k=1,\ldots,Q \tag{7.25}$$

$$\sum_{t=0}^{T}\sum_{i\in D_k} f_{idk}(t) = q_k,\ k=1,2,\cdots,Q \tag{7.26}$$

$$y_i(t+1)-y_i(t)=\sum_{l\in pred(i)}\sum_{k=1}^{Q} f_{lik}(t-\lambda_{li})-\sum_{j\in succ(i)}\sum_{k=1}^{Q} f_{ijk}(t), \quad \forall i\in N, t=0,\ldots,T \tag{7.27}$$

$$y_i(0)=0,\ \forall i\in N, i\neq s \tag{7.28}$$

$$y_i(t)=0,\ \forall i\in D, t=0,\ldots,T \tag{7.29}$$

$$0\le y_i(t)\le a_i, t=1,\ldots,T, \forall i\in N-D \tag{7.30}$$

$$0\le \sum_{k=1}^{Q} f_{ijk}(t)\le x_{ij}^k\left(c_{ij}+z_{ij}^k c_n\right)+\bar{x}_{ji}^k\left(c_{ji}+\bar{z}_{ij}^k c_n\right), \quad t=TET_{k-1},\ldots,TET_k-\lambda_{ij}, \forall (i,j)\in A, k=1,\ldots,Q \tag{7.31}$$

$$x_{si}^k=1, x_{is}^k=0, \forall (s,i)\in A, k=1,\ldots,Q \tag{7.32}$$

$$x_{ij}^k=1, x_{ji}^k=0, \forall (i,j)\in D_k,\ \ k=1,\cdots,Q \tag{7.33}$$

$$x_{ij}^0=1, \forall (i,j)\in A \tag{7.34}$$

In the above, Eq. (7.15) represents the first objective that is to minimize the total evacuation time. Eq. (7.16) represents the second objective to minimize the total number of lanes that need the contraflow operation. Eq. (7.17) represents that TET_k is the total evacuation time of the *k*th category victims. Eq. (7.18) represents the evacuation priorities of victims of different categories. Eq. (7.19) re presents that x_{ij}^k is a 0-1 variable. Eq. (7.20) represents that $\bar{x}_{ij}^k$ is the opposite of x_{ij}^k. Eq. (7.21) represents the design variable z_{ij}^k is a 0-1 variable. Eq. (7.22) represents that $\bar{z}_{ij}^k$ is the opposite of z_{ij}^k. Eq. (7.23) represents the constraint of input resources for design. Eq. (7.24) represents the constraint of input resources for contraflow. Eq. (7.25) represents the numbers of different categories of victims that are needed to be evacuated from the source node. Eq. (7.26) represents that the victims of *k*th category are all evacuated to the destination nodes in D_k . Eq. (7.27) is to calculate how many victims of different categories stay in node i at time t. Eq. (7.28) represents that there are no victims in the nodes, except the source node. Eq. (7.29) represents that there are no victims in the destination nodes, which is because of the use of super destination node. Eq. (7.30) represents the constraint of the node capacity. Eq. (7.31) represents the constraint of the arc capacity during the kth period. Eq. (7.32) represents the flow pattern of the roads associated with the source node s. Eq. (7.33) represents the flow pattern of the roads associated with the sink nodes during the kth period. Eq. (7.34) represents the initial flow patterns.

7.3.3 Solving the Model

7.3.3.1 Analysis of the mathematical model

The model has several features. First, the model is a multi-period optimization model. For a situation with Q categories of victims, the model is a Q-period optimization model. In each period, there are two objective functions. The first objective function is to minimize the total evacuation time, and the second objective function is to minimize the difference between a flow pattern and the last flow pattern. Second, the decision variables in the model are $f_{ijk}(t)$, $\left(x_{ij}^k, x_{ji}^k\right)$ and z_{ij}^k , which are determined through three optimization layers: (1) one layer for z_{ij}^k , (2) one layer for $\left(x_{ij}^k, x_{ji}^k\right)$, and (3) one layer for $f_{ijk}(t)$. Third, it is obvious that in each evacuation period, the first objective function relates to the number of different categories of victims and the flow patterns, and the second objective function relates to the (k-1)*th* and kth flow patterns. The study in Section 6.4 demonstrated that there is more than one flow pattern that achieves the minimum total evacuation time. Therefore, among all the flow patterns that optimize the first objective function, there must be one that has the minimum value of $F_2(k)$. Such a solution is the Pareto optimal solution according to the definition of the Pareto optimal solution. This observation leads to the algorithm for this two-objective model, which is presented in details in the following sections.

7.3.3.2 Sketch of the proposed method

The proposed method to solve the model (i.e., Eqs. (7.15)-(7.34)) is an algorithm with three layers (upper layer algorithm, middle layer algorithm and lower layer algorithm) which are further coupled, as shown in Fig. 7.10. In each round of iteration, the upper layer algorithm will need to call the middle layer algorithm, and the middle layer algorithm will need to call the lower layer algorithm. The lower layer algorithm, called priority-based minimum cost multicommodity flow (P-MCMF) algorithm, is to seek the minimum flow time for multicommodities given a flow pattern. The middle layer algorithm called three-pattern particle swarm optimization (TPSO) is a novel discrete version of PSO for multi-objective optimization which is to find a Pareto optimal flow pattern in a given designed transportation network. The upper layer algorithm is to find the best design result for the imbalanced transportation network, and it takes an iterative algorithm (IA), for the solution space is limited due to the constraint of resources for design. The three layer algorithm will be discussed in detail in the following.

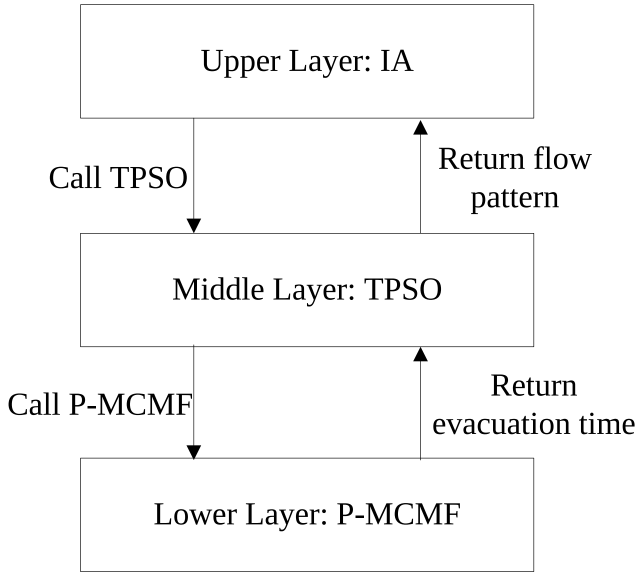

Figure 7.10 Sketch of the three layer algorithm

7.3.3.3 Lower layer algorithm

The lower layer algorithm is designed to solve the evacuation time optimization problem in the model, which is, in essence, a problem of a minimum cost multicommodity flow with priorities. This problem has been studied in the operations research area, and there are many algorithms developed [Wang 2003], for example, basis partitioning method [Farvolden et al 1993], resource-directive method [Geoffrion 1970], price-directive method [Barahona and Anbil 2000], primal-dual method [Polak 1992], approximation method [Fleischer 2000], interior-point method [Goffin et al. 1992], convex programming method [Nagamochi et al. 1990], and heuristics [Barnhart 1993]. However, two issues remain if these algorithms are used for our model: (1) the existing multicommodity minimum cost flow algorithm is designed for static network with no restriction of the node capacity; and (2) the existing minimum cost multicommodity flow algorithm is designed without consideration of priorities.

The first issue can be easily resolved by converting the original traffic network into a time expanded network (*TEN*). For the second issue, because there are different priorities in different categories of victims and the flow pattern can not be changed during one period, the flow pattern optimization has to be taken as a multi-period problem. During one period, the optimization of the evacuation time is similar with the minimum cost flow problem. Nevertheless, we still cannot use any existing minimum cost flow algorithm directly as the lower layer algorithm, for the following reasons: (1) more than one category of victims may need to be evacuated to some safe destinations; and (2) The victims of different categories have different priorities. Therefore, a priority-based minimum cost multicommodity flow (P- MCMF) algorithm is designed as the lower layer algorithm.

The main ideas of the P-MCMF algorithm are: (1) during the *kth* period of the evacuation process, all the *kth* category victims are evacuated as soon as possible, which has the same function as any existing minimum cost flow algorithm, (2) at the same time the k' *th* $(k' = k+1, \cdots, Q)$ categories are evacuated as much as possible. This second idea further means that the transportation network may have an ability to evacuate k' *th* $(k' = k+1, \cdots, Q)$ categories of victims when evacuating *kth* category victim. The detail of the P-MCMF algorithm is expressed in Fig. 7.11.

From the lower layer algorithm we can get TET_k' and $q_{k'}' (k' = k+1, \cdots, Q)$. Here $q_{k'}'$ refers to the number of k' *th* category $(k' = k+1, \cdots, Q)$ victims that have not been evacuated after the *kth* period. It is obvious that $q_k^{'}$ is not larger than q_k , and $TET_k^{'}$ can be expressed by Eq. (7.34).

$$TET_k^{'} = TET_k - TET_{k-1} \tag{7.34}$$

The block "Update *TEN*" has two operations: updating the capacity matrix and updating the destination nodes. It should be noted that MCF-I and MCF-II are different. MCF-I is an existing classic minimum cost flow algorithm, for example the dual ascent algorithm, which is used to evacuate *kth* category victims whose number is $q_k^{'}$ and get the total evacuation time *TET*. MCF-II is to calculate the numbers of victims of $k^{'}$ *th* category $(k^{'} = k+1, \cdots, Q)$ that can be evacuated during the *kth* period. The differences between MCF-I and MCF-II are: (1) there is a vector in MCF-II, labelled as f_{exist} , to record the flows that have been on the transportation network, which means the capacity of the edges decrease; (2) the maximum time in MCF-II, namely the number of time unit of the time expanded network (*TEN*), is fixed which equals to *TET* got by MCF-I, and MCF-II is to calculate the numbers of victims of $k^{'}$ *th* category $(k^{'} = k+1, \cdots, Q)$ that can be evacuated within the maximum time.

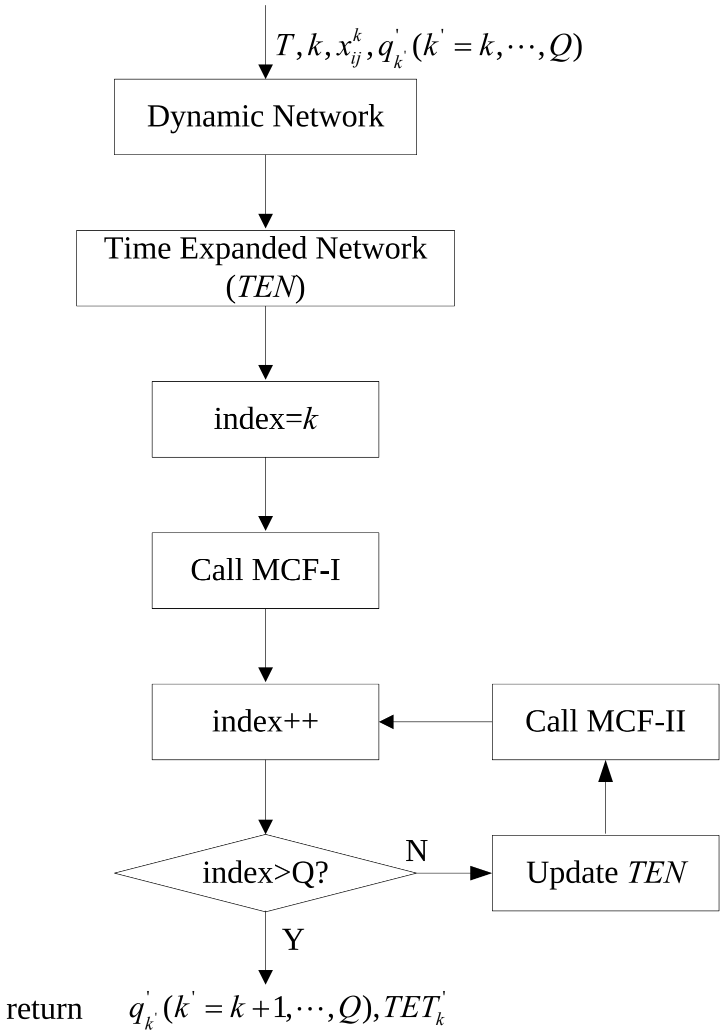

Figure 7.11 Flow chart of P-MCMF algorithm

7.3.3.4 Middle layer algorithm

The middle layer algorithm, namely three-pattern particle swarm optimization (TPSO) algorithm, is used to solve the flow pattern planning problem in the model, which is the key part in the whole method. The TPSO is designed by adding different particle flying patterns and novel fitness evaluation method into the standard PSO, which fits the multiple-objective optimization of our model.

As discussed above, the flow pattern problem is much different from the traditional discrete optimization problem. It is a discrete variable multiple-objective optimization problem; in particular, the first objective has higher priority than the second objective. We design a discrete version three-pattern PSO algorithm by adding the following four measures into the standard PSO: (1) three flying patterns: every particle has three different flying patterns, namely standard pattern, dimension-decreasing pattern and dimension-increasing pattern, respectively; (2) memory of multi-objective information: when flying, the particles consider the two different objectives at the same time, instead of only one objective in the standard PSO, and each particle decides its flying patterns in each iteration according to the two fitness values; (3) synchronous procedure for fitness evaluation: in each iteration, a particle is evaluated immediately after it has been updated, instead of after the whole swarm has been updated; (4) self-adaptive control mechanism for time horizon *T*: time horizon *T*, namely the required total evacuation time, as the upper bound of total travel time, is updated by the latest *TET* got by far, which has important effect on the running time of the algorithm.

The key point of the TPSO is the three particle flying patterns. The standard pattern enables the particles with the exploration ability, i.e. the global searching ability; the dimension-decreasing pattern and dimension-increasing pattern, which in essence are two local searching methods, enable the particles with an exploitation ability. With suitable parameters, the TPSO algorithm will has a good balance between exploration and exploitation.

A problem in the optimization of this model is that the evaluation of the first kind of objective function is a complicated process which needs much more running time than traditional fitness evaluation. The measures (3) and (4) mentioned above, namely the synchronous procedure for fitness evaluation and self-adaptive control mechanism for time horizon *T*, are adapted to solve this problem. The average running time of the algorithm is reduced to a relative low level.

There are four key steps for the implementation of the TPSO and they are: encoding, fitness evaluation, update of the particle swarm, and stop criterion. The encoding step is the same with that in Section 6.4.4.2. The other three steps along with the whole procedure of the TPSO are given as follows.

(1) Fitness evaluation

A. Memory of multi-objective information.

In standard PSO, the best position and the corresponding best fitness value of a particle and those of the whole swam should be recorded, and whether the position the particle is the best is evaluated by only one objective function. But in TPSO, each particle must be evaluated by two objective functions. Among the particles that achieve the minimum value of the first objective function, the one that gets minimum value of the second objective function will be regarded as the best particle of the swarm and represented by symbol *g*. The same rule is taken to decide the best previous position of *i*th particle, which is also recorded as $p_i = (p_{i1}, \cdots, p_{id}, \cdots, p_{iD})$, and the corresponding value of the two objective functions are recorded as f_{1best} and f_{2best}.

B. Synchronous procedure for fitness evaluation.

In standard PSO, the evaluation procedure is as follows. All the particles in the swarm are updated first, and after that all the particles are evaluated. During the evaluation, the best particle so far is indicated with symbol *g*. It is obvious that the best position of swarm get so far will be kept the same during the update process of the whole swarm. This kind of fitness evaluation can be called as asynchronous procedure, because all the particles are updated under the same global best position. But there is a problem here, which is during the update process when a particle, say particle *i*, get better position than p_g, the particle will not be labelled as *g*, for evaluation is not performed. As a result, the new best global position will not affect the update of the particle *i*+1 to particle *m* in this iteration. For a normal optimization problem, there is not much difference here. But for our problem, it is different. The evaluation of the first objective should be as less as possible, because the evaluation of the first objective is an optimization process which will cost much more time than the computation of a regular function. So it will be better that the new best global position affects the update of the other particles immediately after it is obtained.

Based on the reason above, the synchronous evaluation procedure is designed as follows. In each iteration, particle *i* will be evaluated right after it has been updated, and it will be labelled as *g* if it get better position than P_g.

C. Self-adaptive control mechanism for time horizon *T*.

The time horizon *T*, namely required total evacuation time, is used as the bound of total travel time and transferred to P-MCMF by TPSO to get total evacuation time. *T* is used to establish the time expanded network. In our problem, the dynamic network with *X* nodes will result in a static network with $X(T+1)+1$ nodes. Since the running time of P-MCMF is the increasing function of the size of network, the value of *T* is very important for the running time of the whole method. Obviously *T* is much larger than the total evacuation time of higher priorities of victims, which will result in a relative long running time. To solve this problem, we designed a self-adaptive control mechanism for time horizon *T*, which is as follows. In each iteration, *T* is updated by the f_{1best} of the global best position P_g. f_{1best} is the minimum total evacuation time so far, with which we can get the smallest size of the time expanded network.

(2) Update of the particle swarm

Update of the particle swarm, namely the flying patterns of the particles in the searching space, is the key of the TPSO algorithm. Here we will introduce the detail of three particle flying patterns.

A. Standard pattern

The standard pattern implements the basic idea of PSO algorithm and is the same with that in Section 6.4.4.2, which enables a particle with the potential ability to searching the whole solution space.

B. Dimension-decreasing pattern

The dimension-decreasing pattern comes from the following idea. For the particles who have got the best fitness value of the first objective, namely f_{1best} , whether the components of the position vectors resulting in the change of flow pattern are necessary. If not, which means that the number of dimensions of the position vector different with the position corresponding with the latest flow pattern decrease, then the second objective function will get less value. So the dimension-decreasing update pattern is a local searching method, whose purpose is to find the better value of the second objective function based on the current optimal value of the first objective function.

Suppose the position vector related to the flow pattern in last period is represented as X_{last}, then particle i will perform the dimension-decreasing pattern as the procedure in Fig. 7.12.

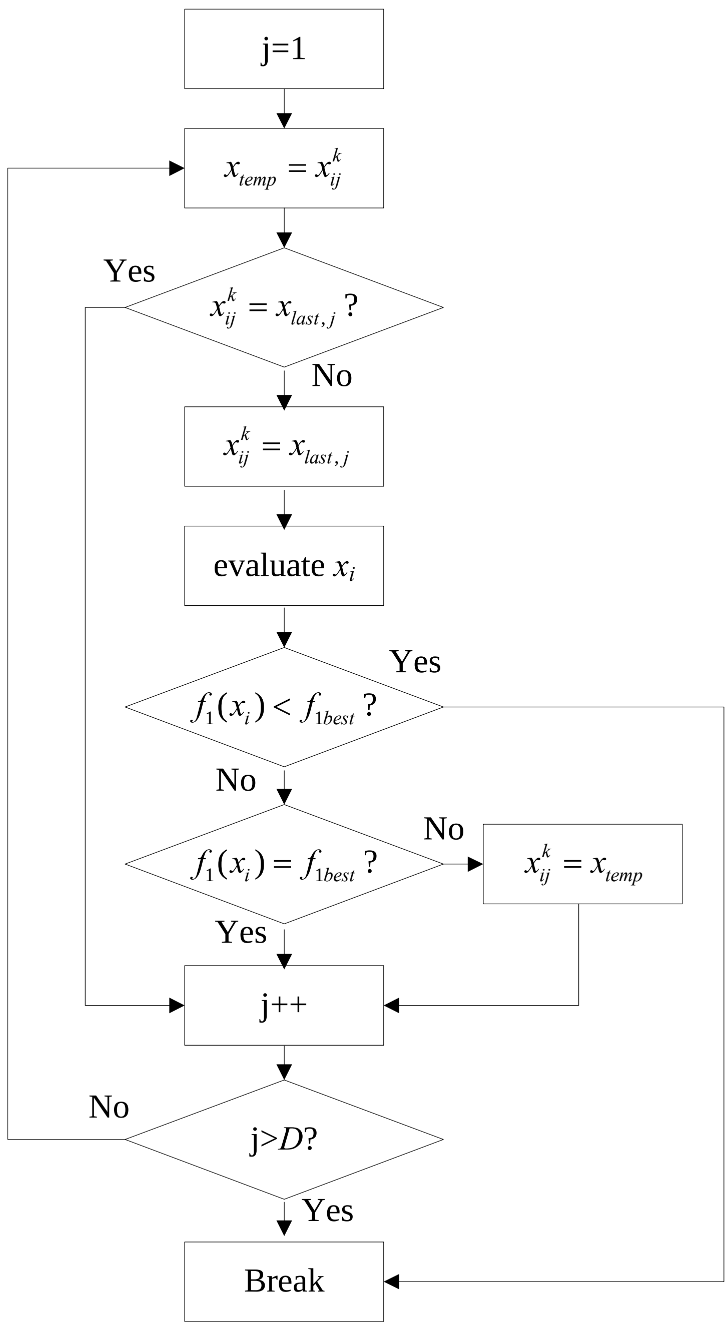

Figure 7.12 Flow chart of dimension-decreasing pattern

The symbol D in Fig. 7.12 represents the number of dimensions of the searching space, namely the component number of the position vector of a particle. It should be noted that, since the dimension-decreasing pattern is to find better value of the second objective function for the particles that have got the f_{1best}, the other particles without f_{1best} will not perform this flying pattern and just stay in its latest position. The purpose of this regulation is to avoid useless fitness evaluations which result in extra running time of the algorithm.

Two parameters, '*dim_de_out*' and '*dim_de_in*', are preset to control the start and the end of the dimension-decreasing pattern. When f_{1best} keeps stable for '*dim_de_out*' iterations, the swarm will update with the dimension-decreasing pattern. After the swarm flying with the dimension-decreasing pattern, when f_{2best}, namely the corresponding second fitness value of the current optimal position vector, keeps stable for '*dim_de_in*' iterations, the swarm will end the dimension-decreasing flying pattern and re-enter the standard flying pattern.

C. Dimension-increasing pattern

The dimension-increasing pattern is the opposite way of the dimension-decreasing. The purpose of dimension-increasing pattern is to find better value of the first objective basing the optimal position so far, through the way of changing the components of the position vector of the current optimal particle which are the same as those of X_{last}, which means that the number of dimensions of the position vector different from X_{last} increases and as a result, the second objective function will get a larger value. The dimension-increasing update pattern is another local searching method. The way of changing the components of the position vector of the current optimal particle also follows the updating formula for the standard pattern.

Our experience from the experiments shows that comparing with the dimension-decreasing method, it is relative more difficult for a particle to find a better f_{1best} with the dimension-increasing pattern than with the dimension-increasing pattern. So in this pattern, all the particles in the swarm will move to the optimal position and perform the dimension-increasing update pattern, which is different from the dimension-decreasing pattern.

Suppose the position vector related to the flow pattern in the last period is represented as X_{last}, and the optimal position vector so far is p_g^k. Then particle i will perform the dimension-increasing pattern as shown in Fig. 7.13.

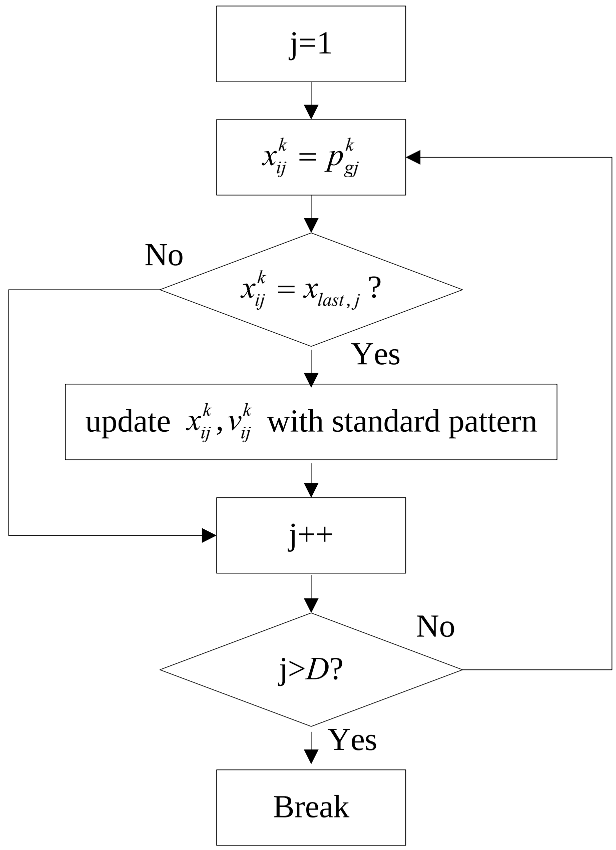

Figure 7.13 Flow chart of dimension-increasing pattern

It is obvious that the particle swarm considers entering the dimension-increasing pattern only after it finishes the dimension-decreasing pattern and has obtained the minimum second fitness value of the current f_{1best}. So three parameters, '*dim_in_out*', '*dim_in_in*' and '*flag*', are used to control the start and the end of the dimension-increasing pattern. '*flag*' is a Boolean variable, which memorizes whether the swarm has experienced the dimension-decreasing pattern under the current f_{1best}. '*dim_in_out*' and '*dim_in_in*' are preset, which perform the similar function with '*dim_de_out*', '*dim_de_in*'. When '*flag*' is true and f_{1best} keeps stable for '*dim_in_out*' iterations, the swarm will enter the dimension-decreasing pattern. After the swarm flying with the dimension-increasing pattern, when f_{2best} keeps stable for '*dim_in_in*' iterations, the swarm will end the dimension-increasing flying pattern and re-enter the standard flying pattern.

(3) Stop criterion

A positive integer *NG* is taken as the maximum iteration number. If the iteration number k is greater than *NG*, the algorithm is stopped. *NG* is set based on the trial-and-error for a particular problem.

(4) Procedure of TPSO

The entire procedure of the algorithm has the following steps.

Step 1: Set *NG*, c_0, c_1, c_2, *dim_de_out*, *dim_de_in*, *dim_in_out* and *dim_in_in*. Initialize a swarm including *m* particles with random positions inside the encoding space. x^* is the best position. f_{1best} is the best first fitness value and f_{2best} is the corresponding second fitness value. Let *Counter1*=0, *Counter2*=0, *Counter3*=0, *Counter4*=0, *flag*=false, *Iter*=0.

Step 2: Call P-MCMF algorithm to evaluate the first fitness of each particle. That the index of the one gets the best first fitness is set as *g*. The position and fitness values of the particle *g* are used to update x^*, f_{1best} and f_{2best}.

Step 3: If *iter*>NG, the algorithm ends; else if *flag*=false, then go to Step 4; if *flag*=true, then go to Step 8.

Step 4: If *Counter1*< *dim_de_out*, then update the swarm with the standard pattern and evaluate the swarm in the synchronous procedure; else go to Step 6.

Step 5: If $f_1(x_g) < f_{1best}$, then let $x^* = x_g, f_{1best} = f_1(x_g), f_{2best} = f_2(x_g)$ and go to Step 3; else *counter1*++ and go to Step 3.

Step 6: If *Counter2*<*dim_de_in*, then update the swarm with the dimension-decreasing pattern, and go to Step 7; else let *flag*=true, *Counter1*=0, *Counter2*=0, and go to Step 3.

Step 7:If $f_1(x_g) < f_{1best}$, then let $x^* = x_g, f_{1best} = f_1(x_g), f_{2best} = f_2(x_g)$, *Counter1*=0, and go to Step 3; if $f_1(x_g) = f_{1best}$ and $f_2(x_g) < f_{2best}$, then let $x^* = x_g, f_{2best} = f_2(x_g)$ and go to Step 3; else *counter2*++ and go to Step 3.

Step 8: If *Counter3*< *dim_in_out*, then update the swarm with the standard pattern and evaluate the swarm in the synchronous procedure; else go to Step 10.

Step 9: If $f_1(x_g) < f_{1best}$, then let $x^* = x_g, f_{1best} = f_1(x_g), f_{2best} = f_2(x_g)$ and go to Step 3; else *counter3*++ and go to Step 3.

Step 10: If *Counter4*<*dim_in_in*, then update the swarm with the dimension-increasing pattern, and go to Step 11; else let *flag*=false, *Counter3*=0, *Counter4*=0, and go to Step 3.

Step 11: If $f_1(x_g) < f_{1best}$ and $r_p f_{2best} \le R_p$, then let $x^* = x_g, f_{1best} = f_1(x_g), f_{2best} = f_2(x_g)$, *Counter3*=0, and go to Step 3; else *counter4*++ and go to Step 3.

7.3.3.5 Upper layer algorithm

The upper layer algorithm is to find the best design result by adding new lanes to some edge in the damaged transportation network. Due to the constraint of input resources for design, usually very limited new lanes can be designed. Therefore, the solution space is limited, and an iterative enumeration algorithm can be taken as the upper layer algorithm, which is illustrated in Fig. 7.14. At the beginning, TET_k are set as the required total evacuation time T. The block "Update and TET_k" is according to the Eq. (7.34).

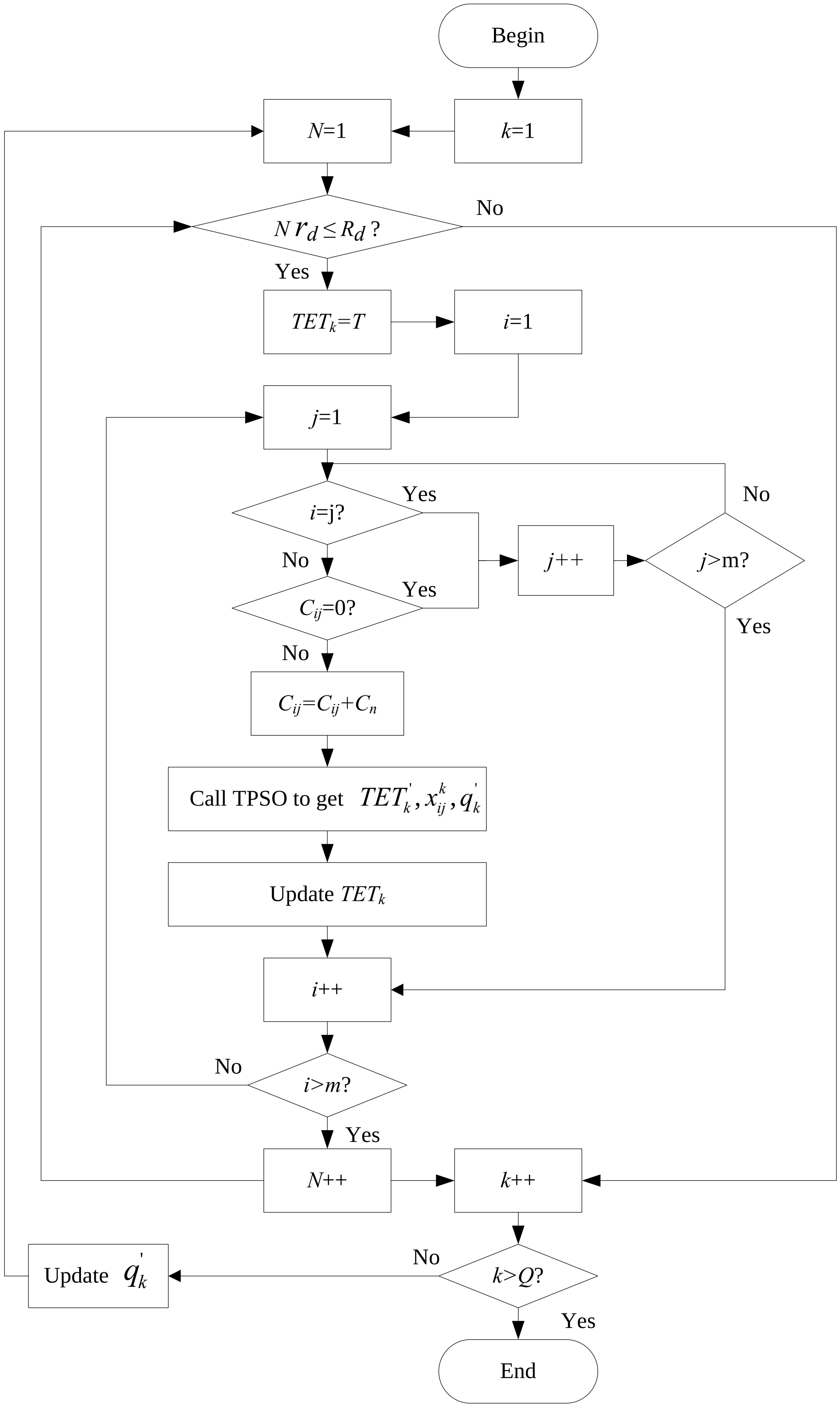

Figure 7.14 Iterative algorithm

7.3.4 Experiment

7.3.4.1 Experimental setting

Fig. 7.15 illustrates an original transportation system which has seven places. The capacity of each place is represented by a bracket "{}"; for example, the capacity of Place 1 is 120 (Fig. 7.15). Further in Fig. 13, the "-" in the bracket adjacent to Place 6 and Place 7 means that Place 6 and 7 are the shelters (i.e., their capacities are unlimited). The road and edge are expressed by connection between places (Fig. 7.15). Both capacity and travel time of the edge are indicted on the arc by a parenthesis "( )"; for example, Edge 1->2 has travel time 2 and capacity 1, and Edge 2->1 has travel time 2 and capacity 2 (Fig. 7.15). The time unit in the example is minute which is omitted in the following paragraphs.

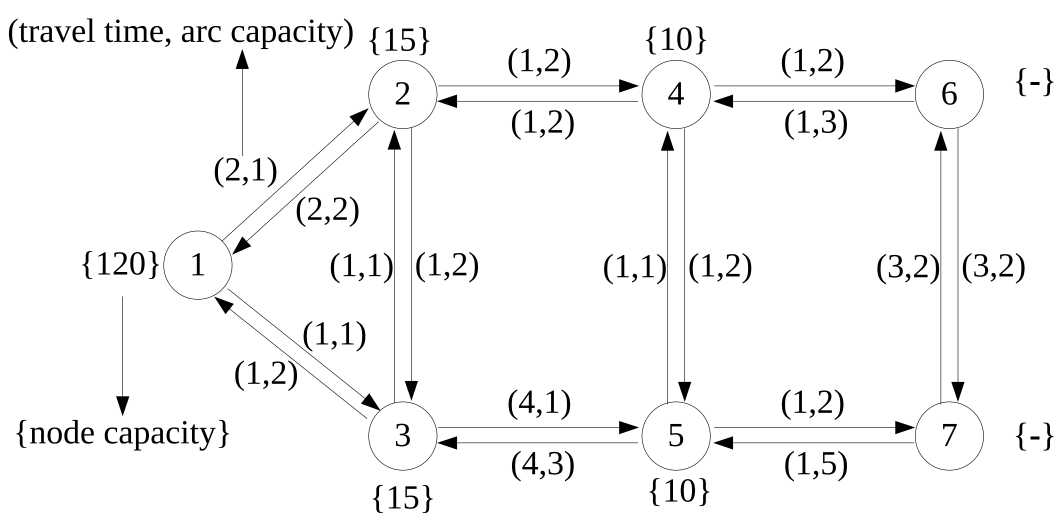

Figure 7.15 The original transportation system

The transportation system is in an imbalanced situation as follows; (1) one edge (4->5) is damaged, and (2) some victims need to be evacuated from a dangerous place to two safe places. As shown in Fig. 7.16, Place 1 is the source node, and Place 6 and 7 are two sink nodes. Place 1 has two categories of victims, the number of the victims in the first category is 15 and the number in the second category is 93. The first category must be evacuated to Place 6 and the second category must be evacuated to Place 7.

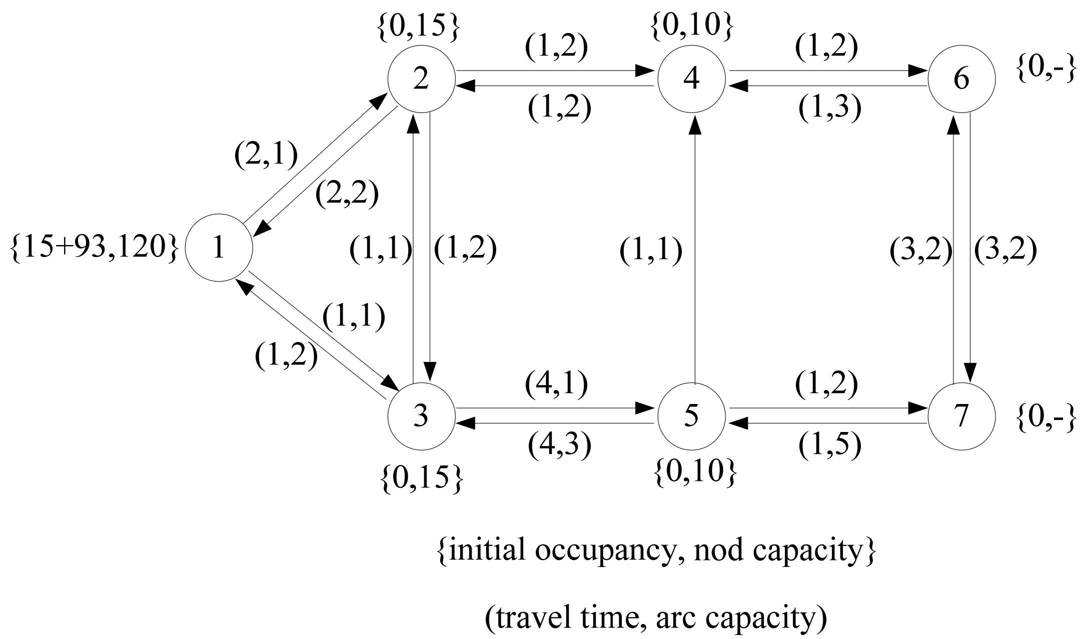

Figure 7.16 Damaged transportation system
(the source node is 1; the destination node for 1st category is 6 and for 2nd category is 7)

There are two categories of input resources are managed for design and flow pattern planning. These resources are further subject to the following constraints.

(1) The resource for design is only enough to add one new lane on the system during one time period and the resource is reusable. The new lane falls into the edge domain of the original transportation system. The travel time of the new lane *i->j* is the same as the original travel time t_{ij} of the edge *i->j* in the original transportation network. The capacity of the new lane is 2.

(2) The resource for flow pattern planning is enough to setup contraflow on 8 roads and the resource is reusable.

Regarding flow pattern planning, according to the constraint equations (7.32) and (7.33), we have already known that the flow patterns for edge (1,2), (1,3), (4,6), (6,7) during the first evacuation period should be ones as shown in Fig. 7.17, and the flow patterns for edge (1,2), (1,3), (5,7), (6,7) during the second evacuation period should be ones as shown in Fig. 7.18. This implies that these edges are no longer a decision variable.

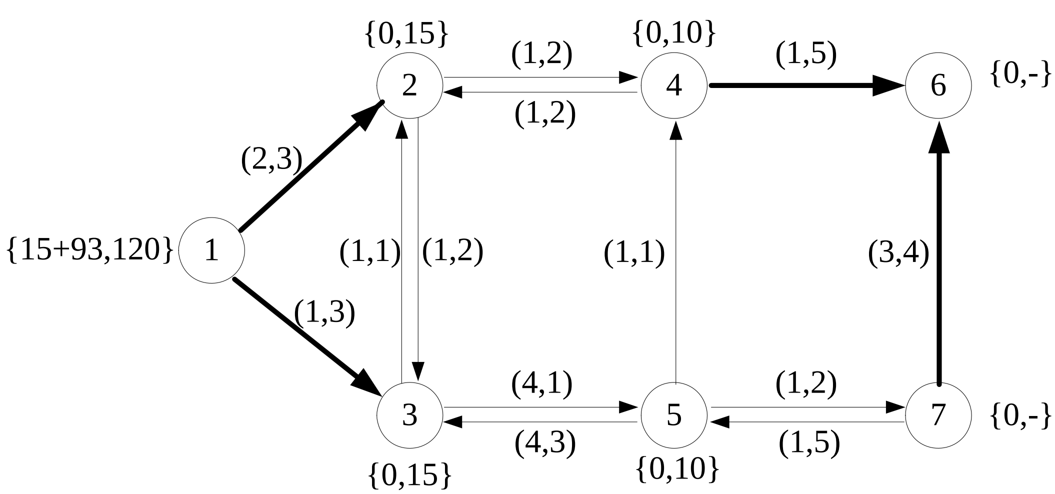

Figure 7.17 Predefined transportation network for category 1

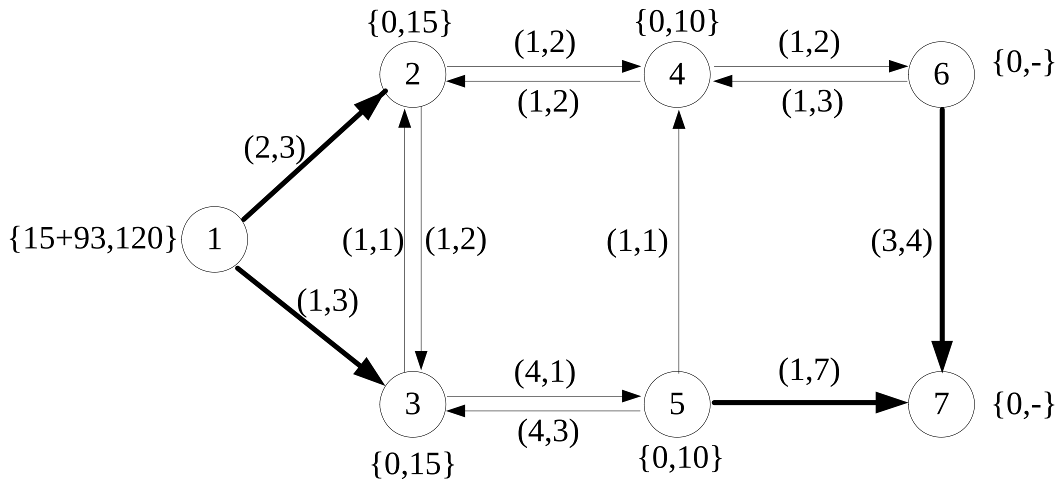

Figure 7.18 Predefined transportation network for category 2

In the experiment, we consider four different rebalancing solutions: (1) damaged transportation network without any reconfiguration strategy, as shown in Fig. 7.16; (2) predefined networks in which the flow patterns of those edges connecting from the source node to the destination nodes are set as those in Fig. 7.17 and Fig. 7.18; (3) transportation network with planning strategy; (4) transportation network with integrated reconfiguration strategy.

In our method to solve this problem, we have the following settings: the swarm size in TPSO, m=5; the maximum iteration number in TPSO, NG=10, the inertia weight in PSO, ω decreases linearly from 0.8 to 0.2. The whole method is programmed with java and the experiment is performed on a computer with a dual-1.66GHz CPU and a 1.5GB memory. The algorithm runs 50 times.

7.3.4.2 Results and discussion

The performance of the 4 rebalancing solutions settings is listed in Table 7.3. The optimal solution with planning strategy for the victims of the 1st category and the 2nd category are shown, respectively, in Fig. 7.19 and Fig. 7.20. With the optimal planning strategy, the total evacuation time of 1st category is 7, and the total evacuation time of 2nd category is 25. Therefore, for both categories of victims, their evacuation performances are much better than those of the damaged network and predefined network.

Table 7.3 Results of different rebalancing solutions

	1st Category TET_1	2nd Category TET_2
Damaged network	11	58
Predefined network	9	43
Flow pattern planning	7	25
Integrated strategy	6	21

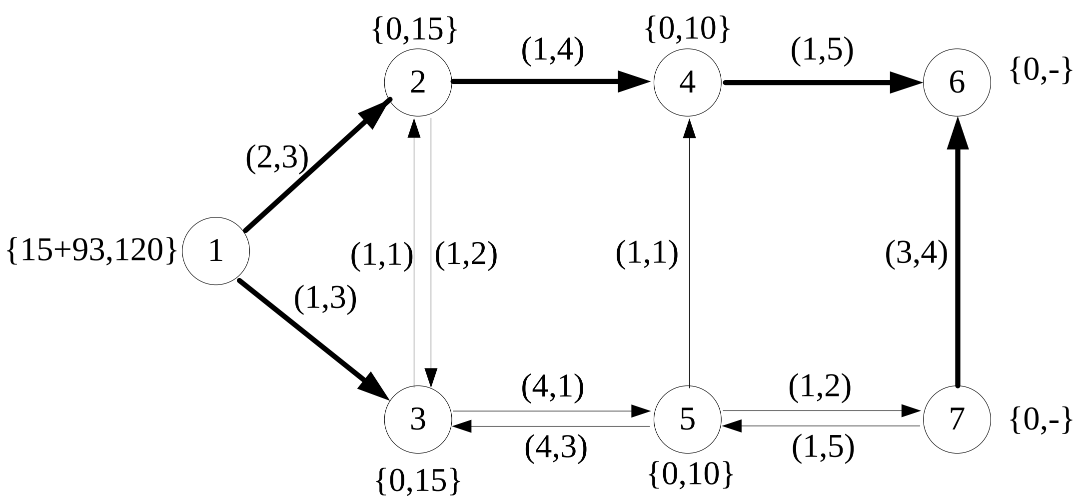

Figure 7.19 The optimal solution with flow pattern planning for the 1st category victim

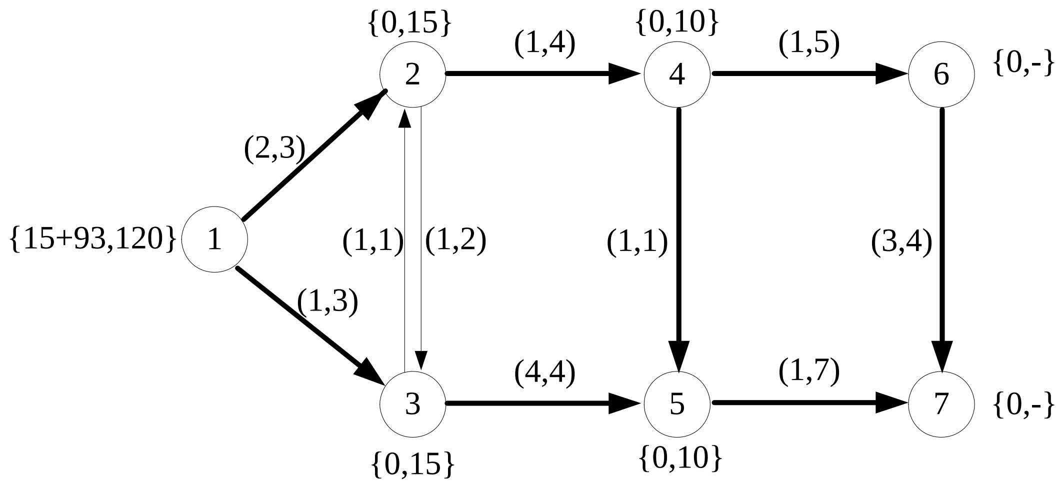

Figure 7.20 The optimal solution with flow pattern planning for the 2nd category victim

For the integrated reconfiguration strategy, during the 1st period the optimal design is found to add a new lane 2->4. The optimal flow pattern under this design is shown in Fig. 7.19. During the 2nd period, the optimal design is found to add a new lane 1->3. The optimal flow pattern under this design is shown in Fig. 7.20. We can see that under this integrated reconfiguration strategy, the total evacuation time of 1st category is 6, and the total evacuation time of 2nd category is 21. Comparing with the results obtained from the damaged network, predefined flow pattern network and network with planning strategy, the rebalancing solution with integrated reconfiguration strategy can reduce the total evacuation time by 60.34%, 46.51%, and 16%, respectively. It should be noted that in an emergency situation the first category victims may be serious ill people and the saving of time means the increasing of the surviving probability.

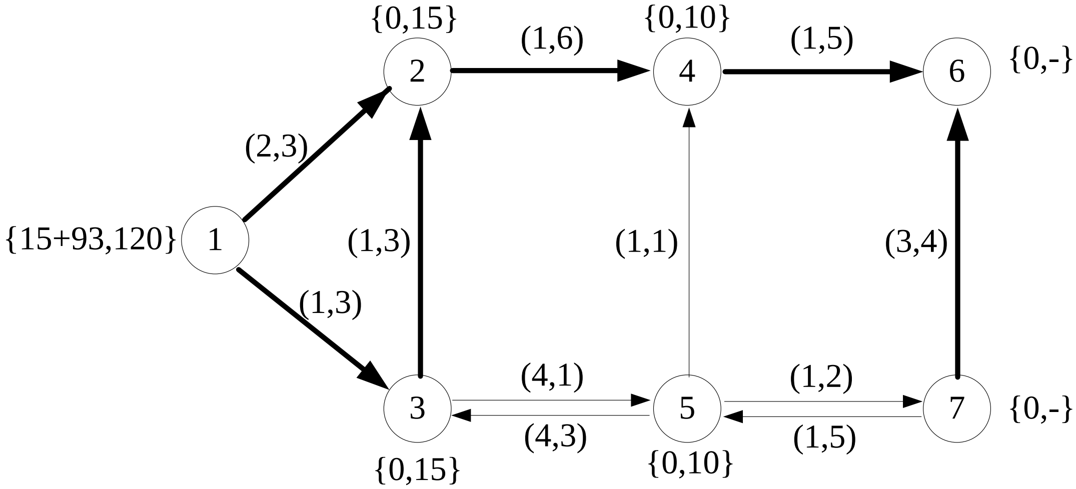

Figure 7.21 The optimal solution with integration strategy for the 1st category victim.

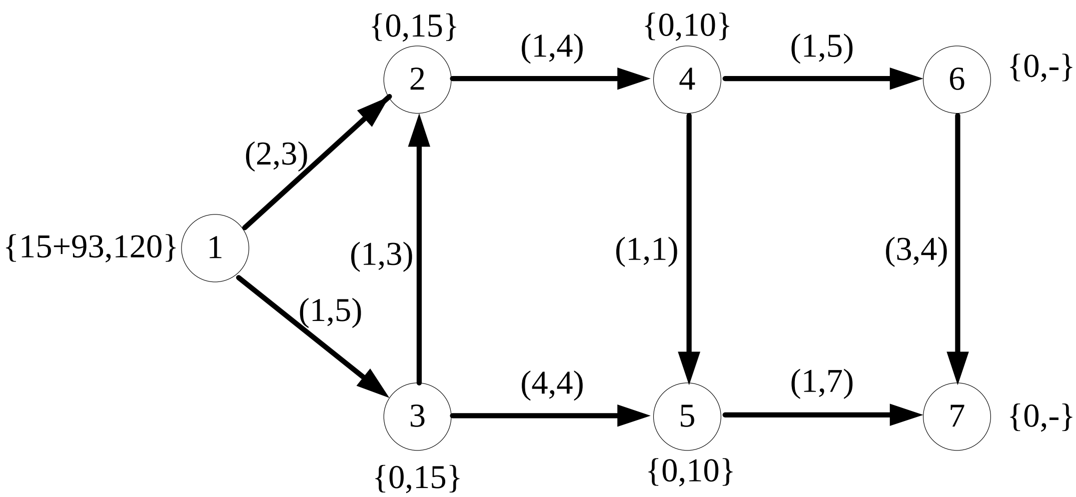

Figure 7.22 The optimal solution with integration strategy for the 2nd category victim.

It is noted that the in the three layer method, the upper layer algorithm and the lower layer algorithm are deterministic algorithms, and the middle layer algorithm, TPSO, is a random searching algorithm, which means that to test the stability and success rate of the method, we only need to test the TPSO algorithm. The best parameters of the TPSO algorithm is as stated in Section 7.3.4.2, under which settings the success rate that PSO achieves the global optimum in the given example is 100%. The running time of the whole method to get design and planning results is about 30 seconds.

To test the scalability issue of the method, we apply the method to the following networks: (i) 11 nodes and 15 two-way arcs; and (ii) 19 nodes and 30 two-way arcs. These two networks are the appropriate size in the real-world application. We test different parameters of TPSO. The algorithm runs 50 times for each case. The parameters of TPSO are much related with the algorithm performance and running time. For example, when the parameters of TPSO for the network with 11 nodes and 15 two-way arcs are set the same as those of network with 7 nodes and 9 arcs, the success rate is 50%. If we increase the swarm size m to 10 and maximum iteration number *NG* to 20, the success rate will increase to 100%. But the good performance results in the running time increasing up to four times the running time with the original parameter setting. The detail results of the TPSO performance with different parameters are listed in Table 7.4. . It should be noted that the average run time of the algorithm is also related to the number of evacuees. The large number of evacuees results in the large size of the time expanded static network for the same dynamic network. For example, there are about 10 time units in the first period of example system in Fig. 7.15, which means that the time expanded network in the first period has about 78 nodes; there are about 30 time units in the second period of case 2, which means that the time expanded network in the second period has about 218 nodes. The difference of the sizes of the two time expanded networks results in significant difference of the average run times.

Table 7.4 Performance of TPSO with different parameters

Example network	Parameters	Success rate	Running time(s)
11 nodes, 15 arcs	m=5,NG=10	50%	48.36
	m=5,NG=20	70%	94.83
	m=10,NG=10	70%	92.67
	m=10,NG=20	100%	183.57
19 nodes, 30 arcs	m=20,NG=5	30%	151.826
	m=20,NG=10	30%	293.894
	m=20,NG=20	80%	571.182
	m=30,NG=5	50%	213.256
	m=30,NG=10	70%	427.938
	m=30,NG=20	90%	864.57

7.3.5 Conclusion

This section uses an example of transportation system to validate the approach of resilience improvement. The result showed that rebalancing solution with integration of design, planning and management can significantly increase the evacuation ability of a transportation system. At the computational aspect, our research concludes that the three layer method is effective for the integrated resource reconfiguration strategy. The upper layer algorithm is to find an optimal design for a damaged transportation system; the TPSO algorithm is to find an optimal flow pattern for the given transportation system; and the lower layer algorithm, P-MCMF algorithm, is to find an optimal flow reconfiguration solution for victims of different categories. The lower-layer algorithm provides an easy but effective solution for a special minimum cost multi-commodity flow (MCMF) problem.

There are some limitations in the rebalancing solution expressed in the model. First, the management strategy in this model considers configuring two categories of resources for design and planning; particularly, the resource for design is used to construct a new lane and the resource for planning is used to setup flow pattern. However, the resources, such as emergency vehicles, which is helpful for planning substance flow has not been considered. In a real evacuation situation, emergency vehicles will also occupy the transportation infrastructure, and consequently, there are two streams of flows on an infrastructure, i.e., a flow of empty vehicles and a flow of victim-loaded vehicles. Second, the upper layer algorithm is designed only for a transportation system; furthermore, it is not suitable for larger amount of resource for design. For a transportation system, the new lanes can be added one by one to the system, which limit the solution space of design problem. For large amount of resource for design, the upper layer algorithm will be very time-consuming.

7.4 Conclusions

This chapter proposed a methodology for improving resilience of a service system by integration of three resource reconfiguration strategies, namely design, planning and management. The framework and mathematical model of the integration strategy are discussed, followed by a validation case of transportation system. The results of the example showed that the integration strategy is an effective resource reconfiguration approach to create rebalancing solutions for an imbalanced service system.

An important assumption of the proposed methodology is that the imbalanced situation faced by the service system has been given. That is to say, the methodology for resilience improvement does not consider uncertainties, which follows the axiom 1 of resilience measurement presented in Chapter 6. This point makes the methodology of resilience improvement is quite different from robustness improvement, as the latter has to consider uncertainties.

The integration of design, planning and management is not only an effective strategy of resilience improvement for a service system; it could be viewed as a general strategy to improve different properties of different systems, as it is a general framework for configuration of different resources. For example, the strategy may be used to improve the entire performance of a manufacturing system by configuring different resources in its different subsystems; the strategy may also be used to improve other safety related properties of a service system, such as reliability and robustness.

CHPATER 8
CONCLUSIONS AND FUTURE WORK

8.1 Overview and Conclusion

This thesis aimed to advance our understanding on service along with service system and its resilience property and/or behaviour. In the current literature, ‘service’ and ‘resilience’ are two buzzwords. However, the concepts of them lacked clarity, which negatively affected their development and application. To have a more thorough understanding of the two and formulate a more sound theory for them was the main motivation of the study presented in this thesis.

The following works have been completed and they were presented in this thesis:

- Review and analysis of service system and resilience engineering;
- Development of the definition of service and service system and of the domain model for service systems;
- Development of the definition of resilience and of the framework for analysis of the resilience of service systems;
- Clarification of the relationships among reliability, robustness and resilience;
- Development the measure and measurement method for the resilience of service systems; and
- Development of the theory and methodology for enhancing the resilience of service systems.

The following conclusions can be drawn from the research results obtained:

(1) The new definition of service and service system enables to distinguish a service system from other systems, e.g., manufacturing systems, etc. That further implies that service system has its unique identity.

(2) The essence of the resilience of a service system is rebalancing between supply and demand.

(3) The measurement of the resilience of a service system makes sense for a particular service system only.

(4) The resilience of a service system is measured by the rebalancing ability from the aspects of time and resource.

(5) The resilience of a service system can be enhanced or improved by not only the service flow planning but also the service infrastructure design and construction.

8.2 Contributions

The contributions of the thesis can be summarized as follows:

Scientifically, this thesis has improved our understanding of service systems and their resilience property or behaviour; furthermore, this thesis has advanced the state of knowledge of safety science in particular having successfully responded to two questions: is a service system safe? how can a service system be made safer?

Technologically and methodologically, the thesis has advanced the state of knowledge for modeling and optimization of complex systems. Generally speaking, this thesis takes a system perspective to view and study the property (resilience) of service systems. Two ideas in this thesis can be generalized and applied to other properties of other systems, namely the integration strategy for the resource configuration and multiple layer optimization models along with the algorithm to solve the model. It is noted that any system relies on its inner and outer resources and configuration of these resources determines a system's overall performance. Thus, the integration strategy for resource configuration is a general strategy for any kind of systems. The computational model for this integration strategy is a multiple layer optimization model. Thus, the multiple layer optimization technique is necessary for a resilient service system.

8.3 Limitations and Future Work

Some limitations remain in this thesis, and they are discussed below.

First, the work reported in this thesis only focused on the resilience of one service system. However, the example of the India power system failure in July 2012 has shown that there are complicated interdependences among different service systems. Therefore, understanding of interdependency among a group of service systems should be improved in future. This may need first of all a model of such a complex system, and then simulation of the model may generate insights on the resilience property or behaviour of the system. A modeling approach based on the function-behaviour-structure framework and agent technology may be suitable to this modeling task.

Second, the proposed approach to enhancing resilience only focused on the technology aspect. The resilience of a system depends on its potential ability to respond to unbalancing situations, and this response ability further depends on reconfiguration of different resources. Obviously, enough resources are important for resilience. However, every resource has a cost, and a system needs to achieve a balance between the resilience and cost. Therefore, it is worthwhile to study the economics of a resilient system. The cost may cover both development cost and operation management cost.

Third, data of large scale practical service systems need to be employed to test the proposed methodologies, especially the proposed algorithms. That is to say, the scalability issue has not been well discussed and should be studied in future. Along with the scalability issue, the efficiency of algorithms needs to be studied in future as well. A so-called approximate algorithm may be useful to such complex systems, compensated by feedback regulation, to achieve an acceptable level of accuracy in management of the resilience of a service system or a group of interdependent service systems.

Fourth, the model of a service system for example transportation system has not captured flow of tools (e.g., vehicles to evacuate victims), which compromises the validity of the model. This is so because the tools also occupy the infrastructure resource. Including both substances (tools and goods) in the model is important to the application of the theoretical developments in this thesis or in future.

References

[1] Allenby B. and Fink J. (2005). Toward inherently secure and resilient societies. Science, 309: 1034-1036.

[2] Almeida, J., Almeida, V., Ardagna, D., Francalanci, C. and Trubian, M. (2006). Resource management in the autonomic service-oriented architecture. In: ICAC '06: IEEE International Conference on Autonomic Computing: 84–92.

[3] Alter, S. (2003). 18 Reasons Why IT-Reliant Work Systems Should Replace the IT Artifact as the Core Subject Matter of the IS Field. Communications of the Association for Information Systems (12:23), pp. 365-394.

[4] Alter, S. (2006). The Work System Method: Connecting People, Processes, and IT for Business Results. Larkspur, CA: Work System Press.

[5] Alter, S. (2008a). Service system fundamentals: work system, value chain, and life cycle. IBM systems journal, 47(1): 71-85.

[6] Alter, S. (2008b). Service system innovation. In IFIP International Federation for Information Processing, Volume 267, Information Technology in the Service Economy: Challenges and Possibilities for the 21st Century, eds. Barrett, M., Davidson, E., Middleton, C., and DeGross, J. (Boston: Springer), pp. 61-80.

[7] Anderson, C. and Bartholdi, J. J. (2000). Centralized versus decentralized control in manufacturing: Lessons from social insects. In: I.P. McCarthy and T. Rakotobe-Joel, eds. Complexity and Complex Systems in Industry, September. UK: The University of Warwick, 92–105.

[8] Barahona F. and Anbil R. (2000). The volume algorithm: producing primal solutions with a subgradient method, Mathematical Programming, 87(3): 385-

399.

[9] Barnhart C. (1993). Dual-ascent methods for large-scale multicommodity flow problems, Naval Research Logistics, 40:305-324.

[10] BBC (2003a). http://news.bbc.co.uk/2/hi/3146136.stm

[11] BBC (2003b). http://news.bbc.co.uk/2/hi/uk_news/england/london/3189755.stm

[12] Bertsekas D. P. and Tseng P. (1994). Relax IV: a fast version of the Relax code for solving minimum cost flow problems. In LIDS-P-2276, MIT, Cambridge, MA.

[13] Billinton R. and Allan R. N. (1996). Reliability evaluation of power systems. Second edition, Plenum Press, New York.

[14] Birolini A. (2010). Reliability engineering: theory and practice, six edition. Springer.

[15] Bongard J., Zykov V. and Lipson H. (2006). Resilient Machines Through Continuous Self-Modeling, Science 314: 1118-1121.

[16] Booth D., Haas H., McCabe F., Newcomer E., Champion M., Ferris C. and Orchard D. (2004). Web Services Architecture. (W3C, Working Draft, 2004; http://www.w3.org/TR/ws-arch/).

[17] Box, D. (2004). Code name indigo: A guide to developing and running connected systems with indigo. MSDN Magazine, http://msdn.microsoft.com/en-us/magazine/cc164026.aspx.

[18] Bruneau M., Chang S. E., Eguchi R. T., Lee G. C., O'Bourke T. D., Reinhorn A. M., Shinozuka M., Tierney K., Wallace W. A. and van Winterfeldt D. (2003). A framework to quantitatively assess and enhance the seismic resilience of

communities. Earthquake spectra, 19(4): 733-752.

[19] Bullinger H. J., Fahnrick K. P. and Meiren T. (2003). Service engineering—methodical development of new service products. Int J Prod Econ 85:275–287.

[20] Bursztein E. and Goubault-Larrecq J. (2007). A logical framework for evaluating network resilience against fault and attacks. Lecture notes in computer science. 4846: 212-227.

[21] Cai Z., Kumar V., Cooper B. F., Eisenhauer G., Schwan K. and Strom R.E. (2006). Utility-driven proactive management of availability in enterprise-scale information flows. Proceedings of the ACM/IFIP/USENIX 2006 International Conference on Middleware, SESSION: Management II: 382-403.

[22] Cao, J., Wang, J., Zhang, S. and Li, M. (2004). A dynamically reconfigurable system based on workflow and service agents. Engineering Applications of Artificial Intelligence. 17(7): 771–782.

[23] Chen H. and Han K. (1994). Improved quantitative measures of robustness for multivariable systems, IEEE transactions on automatic control, 39(4): 807-810.

[24] Cheres E. Palmor Z. J. and Gutman S. (1989). Quantitative measures of robustness for systems including depayed pertubations. IEEE transactions on automatic control, 34(11): 1203-1204.

[25] Cherbakov L., Galambos G., Harishankar R., Kalyana S. and Rackham G. (2005). Impact of Service Orientation at the Business Level, IBM Systems J., 44(4):653-668.

[26] Christopher M. and Peck H. (2004). Building the Resilient Supply Chain, The International Journal of Logistics Management, 15(2):1-14.

[27] CNN (2003). http://articles.cnn.com/2003-08-14/us/power.outage_1_outage-power-plant-lightning-strike?_s=PM:US

[28] Constantin, J. A. and Lusch, R. F. (1994). Understanding Resource Management. The Planning Forum, Oxford, OH.

[29] CTV (2008). http://www.ctv.ca/servlet/ArticleNews/story/CTVNews/20080703/fire_sask_080703/20080703/.

[30] Daskin M.S. (2010). Service science, John Wiley & Sons, Inc.

[31] De Kleer, J. (1984). How circuits work. Artificial Intelligence, Vol. 24, pp: 205-280.

[32] Demirkan H., Spohrer J. C. and Krishna V. (2011). The Science of Service Systems, Service Science: Research and Innovations in the Service Economy. Springer Science+Business Media, LLC.

[33] Demirkan H., Kauffman R. J., Vayghan J.A., Fill H., Karagiannis D. and Maglio P. P. (2008). Service-oriented technology and management: Perspectives on research and practice for the coming decade, Electronics commerce research and applications 7: 356-376.

[34] Dictionary. com (2012). http://dictionary.reference.com/browse/resilience?s=t&ld=1125.

[35] Dinh T. L. and Thi T. T. P. (2010). A conceptual framework for service modeling in a network of service systems, in *Exploring services science*, LNBIP, J-H. Morin, J. Ralyte and M. Snene, Eds, Springer, NY, 2010, pp. 192-206.

[36] DMTF (2005). Common information model (cim) infrastructure specification. Web, accessed on Nov. 15, 2011,

http://www.dmtf.org/standards/published_documents/DSP0004V2.3_final.pdf.

[37] Doltsinis I. and Kang Z. (2004). Robust design of structures using optimization methods. Comput. Methods Appl. Mech. Engrn. 193: 2221-2237.

[38] Dong Z. and Xue D. (1997). Intelligent scheduling of contraflow control operation using hierarchical pattern recognition and constrained optimization., 1997 IEEE International Conference on Systems, Man, and Cybernetics, 1997. “Computational Cybernetics and Simulation”, 12-15 Oct. 1: 135 – 140.

[39] Ericson, C. A. (2011). Concise encyclopedia of system safety: definition of terms and concepts, First Edition, John Wiley & Sons, Inc.

[40] Fairbairn W. (1865). Treadise on iron ship building: its history and progress. London: Longmans, Green and Co.

[41] Farvolden J., Powell W. and Lustig I. (1993). A primal partitioning solution for the arc-chain formulation of a multicommodity network flow problem, Operations Research, 41(4): 669-693.

[42] Feng L. and Brandt R. D. (1998). An optimal control approach to robust control of robot manipulators. IEEE transactions on robotics and automation. 14(1): 69-77.

[43] Ferreira P., Wilson J. R., Ryan B. and Sharples S. (2010). Measuring resilience in the planning of rail engineering work, in Resilience Engineering in practice: a guidebook, edited by Hollnagel E., Paries J., Woods D. and Wreatha J. Ashgate.

[44] Ferry T. S. (1988). Modern accident investigation and analysis. Second Edition, New York, Wiley.

[45] Fisher R. A. (1951). Design of experiments. Edinburgh: Oliver & Boyd.

[46] FirstPost (2012). http://www.firstpost.com/world/why-indias-power-outage-should-ring-alarm-bells-in-the-us-403852.html

[47] Fitzsimmons J. A. and Fitzsimmons M. J. (2004). Service Management: Operations, Strategy, and Information Technology. 4th ed., McGraw-Hill, New York, NY.

[48] Fleischer L. (2000). Approximating fractional multicommodity flow independent of the number of commodities, SIJDM: SIAM Journal on Discrete Mathematics, 24(4): 505-520.

[49] Foster I. (2005). Service-oriented science. Science, 308: 814-817.

[50] Gavrilovska A., Schwan K. and Oleson V. (2002). A practical approach for 'zero' downtime in an operational information system. In: Proc. of the 22nd International Conference on Distributed Computing Systems: 345-352.

[51] Geoffrion A. (1970). Primal resource-directive approaches for optimizing nonlinear decomposable systems, Operations Reasearch, 18: 375-403.

[52] Glushko R. J. and Tabas L. (2009). Designing service systems by bridging the "front stage" and "back stage". Inf Syst E-bus Manage, 7: 407-427.

[53] Goffin J. L., Haurie A. and Philips N. (1992). Decompositon and nondifferentiable optimization with the projective algorithm, Management Science, 38: 284-302.

[54] Gordon J. E. (1949). The epidemiology of accidents, American Journal of public health, 39(4): 503-515.

[55] Graat, E., Midden, C. and Bockholts, P. (1999). Complex Evacuation: Effects of Motivation Level and Slope of Stairs on Emergency Egress Time in a Sports

Stadium, Safety Science, 31 : 127-141.

[56] Gunderson L. H. (2000). Ecological resilience, Annual review of ecological and systematics, 31: 425-439.

[57] Hall-May M. and Surridge M. (2010). Resilient critical infrastructure management using service oriented architecture, International conference on complex, intelligent and software intensive systems, 15-18 Feb. 2010, 1014-1021.

[58] Hamacher H. W. and Tjandra S. A. (2002). Mathematical modelling of evacuation problems: a state of art, In: Pedestriori and Evacuation Dynamics, (Schreckcnberg M. and Sharma S. D., eds., Springer) 227-266.

[59] Hamel G. and Valikangas L. (2003). The quest for resilience. Harvard business review, 81(9): 52-65.

[60] Hawks C. and Reed C. (2006). Theoretical steps towards modeling resilience in complex system. Lecture notes in computer science, 3980: 644-655.

[61] He Y. (2008). A novel approach to emergency management of wireless telecommunication system. Master thesis, Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada.

[62] Heinrich H. W. (1931). Industrial accident prevention. New York: McGraw-Hill.

[63] Holling C. S. (1973). Resilience and stability of ecological systems, Annual Review of Ecology and Systematics, 4: 1-23.

[64] Holling C. S. (1996), Engineering resilience vs. ecological resilience, In Engineering within Ecological Constraints, ed. PC Schulze, pp. 31–43. Washington, DC: Natl.

[65] Holling C. S. (2001). Understanding the complexity of economic, ecological and

social systems, Ecosystems, 4: 390-405.

[66] Holling C. S. and Gunderson L. H. (2002). Resilience and adaptive cycles. In: Panarchy: understanding transformations in human and natural systems, edited by Gunderson L. H. and Holling C. S., page 25-62, Island Press, Washington D. C., USA.

[67] Holling C. S., Gunderson L. H. and Ludwig D. (2002). In quest of a theory of adaptive change, In: Panarchy: understanding transformations in human and natural systems, edited by Gunderson L. H. and Holling C. S., page 3-24, Island Press, Washington D. C., USA.

[68] Hollnagel E. (1998). Cognitive reliability and error analysis method, Oxford, Elservier Science.

[69] Hollnagel E. (2004). Barriers and accident prevention, Hampshire, Ashgate.

[70] Hollnagel E. (2006). Cognitive reliability and error analysis method, http://www.ida.liu.se/~eriho/CREAM_M.htm

[71] Hollnagel E. (2002). Understanding accidents-from root causes to performance variability, IEEE 7th human factors meeting, Scottsdale Arizona.

[72] Hollnagel, E., Goteman O. (2004). The Functional Resonance Accident Model, Proceedings of Cognitive System Engineering in Process Plant 2004 (CSEPC 2004), pp. 155-161.

[73] Hollnagel E., Paries J., Woods D. and Wreathall J. (2011). Resilience engineering in practice: a guide book. Ashgate.

[74] Hollnagel E. and Woods, D. D. (1983). Cognitive systems engineering: new wine in new bottles, Int. J. Man-Machine Studies (18): 583-600.

[75] Hollnagel E., Woods, D. D. and Leveson N. (2006). Resilience engineering concepts and precepts. Ashgate.

[76] IBM (2011a). Services computing. Retrieved on Nov. 15, 2011, from: https://researcher.ibm.com/researcher/view_pic.php?id=152.

[77] IBM (2011b). Services Sciences, Management and Engineering. Retrieved on Nov. 12, 2011 from:http://researchweb.watson.ibm.com/ssme/services.shtml

[78] IfM & IBM. (2008). Succeeding through Service Innovation: A Service Perspective for Education, Research, Business and Government. Cambridge, United Kingdom: University of Cambridge Institute for Manufacturing. ISBN: 978-1-902546-65-0. http://www.ifm.eng.cam.ac.uk/ssme/documents/080428ssi_us_letter.pdf.

[79] India today online (2012). http://indiatoday.intoday.in/story/massive-power-outage-hits-delhi-again-halts-metro-services/1/211127.html

[80] Jameela A. and Mohamed N. (2011). Service-oriented middleware: a survey. Journal of network and computer applications, in press.

[81] Jones S. (2005). Towards an acceptable definition of service. IEEE software, 22(3): 87-93.

[82] Kanno T., Fujii T., Watari R. and Furuta K. (2011). Modeling Dialysis Service in a Disaster to Assess Its Resilience, Proc. 4th. Symp. Resilience Engineering, Sophia Antipolis, France.

[83] Kaplan, C. P., Turner, S., Norman, E. and Stillson, K. (1996). Promoting resilience strategies: A modified consultation model. Social Work in Education, 18(3), 158-168.

[84] Karle T. and Oberweis A. (2008). Collaborative model driven software development for SOA-based systems, in: International United Information Systems Conference (UNISCON 2008), Klagenfurt, Austria, 2008, pp. 189–200.

[85] Karni R. and Kaner M. (2006). An engineering tool for the conceptual design of service systems, In Advances in Service Innovations, edited by Spath and Fahnrich. Springer. NY, 66-83.

[86] Keiller P. A. (1991). On the use and the performance of software reliability growth models, Reliability engineering and system safety, 32: 95-117.

[87] Kennedy J. (1997). The particle swarm: social adaptation of knowledge. In: Proc IEEE Int Conf on Evolutionary Computation. USA, IEEE Service Center, 303-308.

[88] Kennedy J. and Eberhart R. C. (1995). Particle swarm optimization. In: Proc IEEE Int Conf on Neural Networks, Perth, Australia, 1942-1948.

[89] Kennedy J. and Eberhart R. C. (1997). A discrete binary of the particle swarm algorithm. Proc. IEEE International Conference on systems, man and cybernetics, Orlando, pp.4104-4108.

[90] Kim S. and Shekhar S. (2005). Contraflow network reconfiguration for evacuation planning a summary of results, GIS'05, in Bremen, Germany, November 4-5, 250-259.

[91] Kim S., Shekhar S. and Min M. (2008). Contraflow transportation network reconfiguration for evacuation route planning, IEEE transactions on knowledge and data engineering, 20(8): 1115-1129.

[92] Kontogiannis K., Lewis G.A., Smith D.B., Litoiu M., Muller H., Schuster S. and

Stroulia E. (2007). The Landscape of Service-Oriented Systems: A Research Perspective. International workshop on systems development in SOA environments: 1-6.

[93] Krishnamurthy A. (2007). From Just in Time Manufacturing to On-Demand Services, Integrated Series In Information Systems, 1, Volume 16, Service Enterprise Integration, eds by Hsu C. Pages 1-37, Springer.

[94] Kruchten P. (2005). Casting software design in the function-behavior-structure framework. IEEE Software,Vol. 22(2), pp: 52 – 58.

[95] Kuipers B. (1984). Commonsense Reasoning about causality: deriving behavior from structure. Artificial Intelligence, Vol. 24, pp. 169-203.

[96] Lane S. and Richardson I. (2011). Process models for service-based applications: a systematic literature review. Information and software technology. 53: 424-439.

[97] Laprie J.C. (2008). From dependability to resilience. In: International Conference on Dependable Systems and Networks (DSN 2008).

[98] Lau T., Wang H. and Chuang C. (2011). A definition of service as a base for developing service science. International conference on service sciences.

[99] Leveson N. G. (2004a). A new accident model for engineering safer systems, Safety science, 42: 237-270.

[100] Leveson N. G. (2004b). A systems-theoretic approach to safety in software-intensive systems, IEEE transactions on dependable and secure computing, 1(1): 66-86.

[101] Leveson N. G. (2011). Applying systems thinking to analyze and learn from events, Safety science, 49: 55-64.

[102] Leveson N. G. (2000). System safety in computer-controlled automotive system, SAE Congress, March, 2000, available at: http://sunnyday.mit.edu/papers/sae.pdf.

[103] Levitt T. (1972). Production-line approach to services. Harv Bus Rev 50 (Sep. to Oct.): 41-52.

[104] Lewis D., O'Donnell T., Feeney K., Brady A. and Wade V. (2004). Managing user-centric adaptive services for pervasive computing. In: ICAC 2004: 1st International Conference on Autonomic Computing, New York, NY, USA, IEEE Computer Society: 248–255.

[105] Li J.X. (2004). An internet-based configuration design system using constraint-based programming techniques. M. Sc. thesis. University of Saskatchewan, Canada.

[106] Li H., Wang J., Yu H. and Yang D. (2007). Service science, management, and engineering: a literature review in the perspective of management science. IEEE international conference on industrial engineering and engineering management. 1438-1441.

[107] Lin Y. (2007). Reliability evaluation for an information network with node failure under cost constraint. IEEE transactions on systems, man and cybernetics - part A: systems and humans, 37(2): 180-188.

[108] Lin Y. and Yeh C. (2010). Evaluation of optimal network reliability under components-assignments subject to a transmission budget. IEEE transactions on reliability, 59(3): 539-550.

[109] Lin Y. and Zhang W. J. (2004). Towards a novel interface design framework: function-behavior-state paradigm. International journal of human-computer

studies, 2004, 61(3): 259-297.

[110] Lin, Y., and Zhang, W. J. (2005). A function-behavior-state approach to designing human machine interface for nuclear power plant operators. IEEE Transactions on Nuclear Science, Vol. 52(1), pp: 430-439.

[111] Little R. G. (2003). Toward more robust infrastructure: observations on improving the resilience and reliability of critical systems, Proceedings of the 36th Annual Hawaii International Conference on System Sciences.

[112] Liu D. and Deters. R. (2008). Management of service-oriented systems. Service oriented computing and applications, 2: 51-64.

[113] Liu D., Deters. R. and Zhang W. J. (2010). Architectural design for resilience, Enterprise information systems, 4(2): 137-152.

[114] Lovelock C., Wirtz J., and Chew P. (2009). Essential of Services. Marketing. Prentice Hall, Pearson Education South Asia Pte Ltd. Singapore.

[115] Løvås G. G. (1998). Models of Wayfinding in Emergency Evacuations, European Journal of Operation Research, 105: 371-389.

[116] Lundberg J., Rollenhagen C. and Hollnagel E. (2009). What-you-look-for-is-what-you-find-the consequences of underlying accident models in eight accident investigation manuals, Safety Science, 47: 1297-1311.

[117] Lusch R. F. and Vargo S. L. (2006). The service-dominant logic of marketing: dialog, debate, and directions. Editors, Sharpe M.E., Armonk, NY.

[118] Luthar S. S., Cicchetti D., & Becker B. (2000). The construct of resilience: A critical evaluation and guidelines for future work. *Child*

[119] Luthans F., Vogelgesang G. R. and Lester P. B. (2006). Developing the

psychological capital of resiliency. Human resource development review, 5(1): 25.

[120] Lyu M . R. (1995). Handbook of software reliability engineering McGraw-Hill publishing, 1995, ISBN 0-07-039400-8. Available at: http://portal.research.bell-labs.com/orgs/ssr/book/reliability/introduction.html

[121] Maglio P. P., Kreulen J., Srinivasan S. and Spohrer J. (2006). Service systems, service scientists, SSME, and innovation. Commun ACM, 49(7):81–85.

[122] Mallet R. (1856). On the physical conditions involved in the construction of artillery: an investigation of the relative and absolute values of the materials principally employed and of some hitherto unexpected causes of the destruction of the canon in service. London: Longman, Brown, Green, Longmans and Roberts.

[123] Mallet R. (1862). The first principles of observational seismology as developed in the report to the royal society of London to investigate the circumstances of the great earthquake of December 1957. London: Chapman and Hall.

[124] McAslan A. (2010). The concept of resilience: understanding its origins, meaning and utility. Available at: http://www.torrensresilience.org/images/pdfs/resilience%20origins%20and%20utility.pdf.

[125] Manderbaum A. (1998). Service engineering: modelling, analysis and inference of stochastic service networks. Israel institute of technology, Haifa.

[126] Marco P., Zhang W.J. and Ip W.H., (2010). Robust Management of Virtual Enterprise. ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Montreal, Canada, August 22-26.

[127] Mascio R.D. (2003). Service process control: a method to compare dynamic robustness of alternative service processes, Journal of process control, 13: 645-653.

[128] Marais K., Dulac N. and Leveson N. (2004). Beyond normal accidents and high reliability organizations: the need for an alternative approach to safety in complex systems, Presented at the Engineering Systems Division Symposium, MIT, Cambridge, MA March 29–31, available at: http://klabs.org/DEI/lessons_learned/papers/marais-b.pdf.

[129] McAslan A. (2010). The concept of resilience understanding its origins, meaning and utility. Available at: http://www.torrensresilience.org/images/pdfs/resilience%20origins%20and%20utility.pdf.

[130] McCubbin, H. I. and McCubbin M. A. (1988). Typologies of resilient families: Emerging roles of social class and ethnicity. Family Relations, 37: 247-254.

[131] Merriam Webster (2012). http://www.merriam-webster.com/dictionary/resilience?show=0&t=1355042042.

[132] Metronews (2012). http://metronews.ca/news/calgary/294485/shaw-to-review-service-failure-after-calgary-explosion/#disqus_thread.

[133] Mills P. K. and Moberg D. J. (1982). Perspectives on the technology of service operations. Acad Manage Rev 7(3): 467-478.

[134] Mont O.K. (2002). Clarify the concept of product-service system. Journal of cleaner production, 10: 237-245.

[135] Modarres M., Kaminskiy M. and Krivtsov V. (2010). Reliability engineering and

risk analysis: a practical guide. 2nd edition, Taylor&Francis Group.

[136] Muller N. J.(1999). Managing service level agreements. Int. J. Netw. Manag. 9(3): 155–166.

[137] Muddada R. (2010). Towards resilient supply chain networks. Master thesis, University of Saskatchewan.

[138] Nagamochi H., Fukushima M. and Ibaraki T. (1990), Relaxation methods for the strictly convex multicommodity flow problem with capacity constraints on individual commodities, Network, 20: 409-426.

[139] Nagdi K. (1993). Rubber As Engineering Material: Guideline For Users, Hanser Publisher, Munich.

[140] Najjar W. and Gaudiot J. (1990). Network resilience: a measure of network fault tolerance. IEEE transactions on computer, 39(2): 174-181.

[141] OASIS (2006): Reference model for service oriented architecture v 1.0. Web. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf.

[142] Ordanini A. and Pasini P. (2008). Service co-production and value co-creation: the case for a service-oriented architectute (SOA), European Management Journal, 26(5): 289-297.

[143] Oxford Dictionary (2012). http://oxforddictionaries.com/definition/english/resilience.

[144] Papazoglou M. P. and Heuvel W. V. D. (2006). Service-oriented design and development methodology, International Journal of Web Engineering and Technology, 2 (2006) 412–442.

[145] Papazoglou M. P., Traverso P., Dustdar S. and Leymann F. (2007). Service-

oriented computing: state of the art and research challeges. Computer, 40(11): 38-45.

[146] Paton D., Smith L. and Violanti J. (2000). Disaster response: risk, vulnerability and resilience. Disaster prevention and management, 9(3): 173-180.

[147] Patternson E. S., Woods D. D., Cook R. I. and Render M. L. (2007). Collaborative cross-checking to enhance resilience. Cogn. Tech. Work, 9: 155-162.

[148] Perneger T. V. (2005). The swiss chess model of safety incidents: are there holes in the metaphor? BMC Health Service Research, 5:71.

[149] Perrow C. (1984). Normal accidents: living with high-risk technologies. Basic Book, New York.

[150] Perrow C. (1999). Normal Accidents: Living with High-Risk Technologies. Princeton University Press, 1999.

[151] Phadke M. S. (1989). Quality engineering using robust design. Englewood Clifffs, NJ: Prentice-Hall.

[152] Phaneuf M. (2003). Resilience and nursing, Availbale at: http://www.infiressources.ca/fer/Depotdocument_anglais/Resilience_and_nursing.pdf.

[153] Pinhanez C. (2009). "Human inside" as the key characteristic of service systems. Proc. of QUIS'11, Wolfsburg, Germany. June 2009. Available: http://pinhanez.com/claudio/publications/quis11.pdf

[154] Polak G. (1992). On a parametric shortest path problem from primal-dual muticommodity network optimization, Network, 22: 283-295.

[155] Qureshi Z. H., Ashraf M. A. and Amer Y. (2007). Modeling industrial safety: a sociotechnical systems perspective, A review of accident modelling approaches for complex soci-technical systems, 2007 IEEE International Conference on Industrial Engineering and Engineering Management, Singapore, 1883-1887.

[156] Rai A. and Sambamurthy V. (2006). Editorial notes—the growth of interest in services management: opportunities for information systems scholars, Information Systems Research, 17(4): 327–331.

[157] Rasmussen J. (1997). Risk management in a dynamic society: a modelling problem. Safety Science, 27 (2/3): 183-213.

[158] Reason J. (1990). Human error, Cambridge, UK, Cambridge University Press.

[159] Reason J. T., Carthey J. and de Leval M. R.(2001). Diagnosing "vulnerable system syndrome": an essential prerequisite to effective risk management. Quality Health Care, 10(Suppl II):ii21-ii25.

[160] Redman C. L. (2005). Resilience theory in archaeology. American anthropologist, 107(1): 70-77.

[161] Redman C. L. and Kinzig A. P. (2003). Resilience of past landscapes: resilience theory, society, and the longue duree. Conservation ecology 7(1): 14. Available at: http://www.ecologyandsociety.org/vol7/iss1/art14/print.pdf.

[162] Rigby D. and Cáceres D. (2001). Organic farming and the sustainability of agricultural systems. Agricultural systems, 68(1): 21-40.

[163] Riordan J. (1962). Stochastic service systems. Wiley, New York.

[164] Saleh J. H. and Marais K. (2006). Highlights from early (and pre-) history of reliability engineering, Reliability engineering and system safety, 91: 249-256.

[165] Sampson, S. E. (2001). Understanding Service Businesses, John Wiley & Sons.

[166] Sampson, S. E. and Froehle C. M. (2006). Foundations and implications of a proposed unified services theory. Production and operations management, 15(2): 329-343.

[167] Schmidt M., Hutchison B., Lambros P. and Phippen R. (2005). The Enterprise Service Bus: Making service-oriented architecture real, IBM systems Journal, 44(4): 781-797.

[168] Sengupta K., Heiser D. R., and Cook L. S. (2006). Manufacturing and service supply chain performance: a comparative analysis," The Journal of Supply Chain Management, fall, pp. 4-15.

[169] Serwy R. D. and Ratanen E. M. (2007). Evaluation of a software implementation of the cognitive reliability and error analysis method, Proceedings of the human factors and ergonomics society 51st annual meeting.

[170] Sheffi Y. (2005). Building a resilient supply chain. Harvard business review supply chain strategy, 1(5): 1-11.

[171] Shi Y. and Eberhart R.C. (1998). A modified particle swarm optimizer. In: Proc IEEE Int Conf on Evolutionary Computation. USA, IEEE Service Center, 69-73.

[172] Shimomura Y. and Arai T. (2004). Proposal of service CAD system- a tool for service engineering, CIRP Annals-Manufacturing Technology, 53(1): 397-400.

[173] Sinonet (2012). http://www.sinonet.org/news/china/2012-07-27/216300_2.html.

[174] Smith J. S. (2003). Survey on the use of simulation for manufacturing system design and operation. Journal of manufacturing systems, 22(2): 157-171.

[175] Spohrer J. and Maglio P. P. (2008). The emergence of service science: Toward

systematic service innovations to accelerate co-creation of value. Production and Operations Management 17(3), 1–9.

[176] Spohrer J., Maglio P. P., Bailey J. and Gruhl D. (2007). Steps toward a science of service system, IEEE Computer, 40(1):71-77.

[177] Spohrer J., Vargo S. L., Caswell N. and Maglio P. P. (2008). The service system is the basic abstraction of service science. In Proceedings of the 41st Annual Hawaii International Conference on System Science, January, p. 1-10.

[178] Stanicek Z. and Winkler M. (2010). Service Systems through the Prism of Conceptual Modeling, Service Science, 2(1/2): 112-125.

[179] Svenson O. (1991). The accident evolution and barrier function (AEB) model applied to accident analysis in the processing industries, Risk Analysis, 11(3): 499-507.

[180] Svenson O. (2001). Accident and incident analysis based on the accident evolution and barrier function (AEB) model. Cognitive, Technology & Work, 3(1): 42-52.

[181] Taguchi G. (1986). Quality engineering through design optimization. White Plains, NY: Krauss International Publications.

[182] The open group (2011). Service-oriented architecture (SOA). Retrieved on Nov. 12, 2011 from: http://www.opengroup.org/projects/soa/page.tpl?CALLER=newpage.tpl&ggid=1575.

[183] Tien J. M. and Berg D. (2003), A case for service systems engineering. Journal of systems science and system engineering, 12(1): 13-38.

[184] Tizghadam A. and Leon-Garcia A. (2009). On robust network planning -2009, IEEE, 7th International Workshop on Design of Reliable Communication Networks, Washington, DC.

[185] Tosic V., Pagurek B., Patel K., Esfandiari B. and Ma W. (2005). Management applications of the web service offerings language (wsol). Information Systems 30(7): 564–586.

[186] Trelea I. C. (2003). The particle swarm optimization algorithm: convergence analysis and parameter selection. Information Processing Letters, 85: 317-325.

[187] Tredgold T. (1818a). On the resilience of materials; with experiments, Philosophical magazine series 1, 51(240): 276-279.

[188] Tredgold T. (1818b). On the transverse strength and resilience of timber, Philosophical magazine series 1, 51(239): 214-216.

[189] Tsai W. T. (2005). Service-oriented systems engineering: a new paradigm. Proceedings of the 2005 IEEE International Workshop on Service-Oriented System Engineering: 3-6.

[190] Tseng J. C. and Wu C. H. (2007). An expert system approach to improving stability and reliability of web service. Expert Systems with Applications 33(2): 379–388.

[191] Turner M., Budgen D., and Brereton P. (2003). Turning software into a service, Computer, 36(10): 38-44.

[192] Ulieru M. (2007). Design for resilience of networked critical infrastructure, Inaugural IEEE International Conference on Digital EcoSystems and Technologies, 21-23 Feb. 2007, 540-545.

[193] Ulrich K. T., and Seering W. P. (1988). Function sharing in mechanical design. The Seventh National Conference on Artificial Intelligence, Vol. 1, pp: 342-370.

[194] Umeda Y., Kondoh S., Shimomura Y., and Tomiyama T. (2005). Development of design methodology for upgradable products based on function–behaviour–state modeling. Artificial Intelligence for Engineering Design, Analysis and Manufacturing, Vol. 19(03), pp: 161-182.

[195] Umeda Y., Takeda H., Tomiyama T., and Yoshikawa H. (1990). Function, behaviour, and structure. In Applications of Artificial Intelligence in Engineering, V, (Gero, J.S.,Ed.), Computational Mechanics Publications and Springer-Verlag, pp: 177-193.

[196] Umeda Y. and Tomiyama T. (1995). FBS modeling: Modeling scheme of function for conceptual design. Workshop on Qualitative Reasoning, pp: 271–278.

[197] Van Breda A. D. (2001). Resilience theory: a literature review. Technical report for South African Military Health Service, Military Psychological Institute, Social Work Research & Development. Available at: http://vanbreda.org/adrian/resilience/resilience_theory_review.pdf.

[198] Vargo S. L. and Lusch R. F. (2004). Evolving to a new dominant logic for marketing. Journal of Marketing 68: 1–17.

[199] Vargo S. L. and Lusch R. F. (2006). Service-dominant logic: What It Is, What It Is Not, What It Might Be. In The Service-Dominant Logic of Marketing: Dialog, Debate and Directions, (eds) R. F. Lusch and S. L. Vargo, pp. 43–56. M.E. Sharpe Inc., Armonk.

[200] Vargo S. L., Maglio P. P. and Akaka M. A. (2008). On value and value co-creation: a service systems and service logic perspective. European management journal, 26: 145-152.

[201] Verma A. K., Ajit S. and Karanki D. R. (2010). Reliability and safety engineering, Springer-Verlag London Limited.

[202] Walker B., Carpenter S., Anderies J., Abel N., Cumming G. S., Janssen M., Lebel L., Norberg J., Peterson G. D. and Pritchard R. (2002). Resilience management in social-ecological systems: a working hypothesis for a participatory approach. Conservation ecology, (1): 14.

[203] Wan F.C., Wang D. W. and Li Y. P. (2004). Particle swarm optimization of correlative product combinatorial introduction model, Control and Decision, 19(5): 520-524.

[204] Wang I. L. (2003). Shortest paths and multicommodity network flows, PhD dissertation, Georgia Institute of Technology.

[205] Wang J. W. (2010). Towards Robust and Resilient Networked Service Systems. PhD research proposal report.

[206] Wang J. W., Furuta K. and Zhang W. J. (2013a). "Resilience of a networked service system: its definition and measurement". The Ninth International Conference on Networking and Services, March 24 - 29, 2013 - Lisbon, Portugal.

[207] Wang J. W., Gao F. and Ip W.H. (2010a). Measurement of resilience and its application to enterprise information systems. Enterprise information systems, 4(2): 215-223.

[208] Wang J. W., Ip W. H. and Zhang W. J. (2010b). An integrated Road Construction

and Resource planning approach to the evacuation of victims from single source to multiple destinations, IEEE transactions on intelligent transportation system, 11(2): 277-289.

[209] Wang J. W., Wang H. F, Zhang W. J. and Furuta K. (2012a). On an Integrated Approach to Emergency Evacuation and Resilience for Transportation Service Systems. Submitted to IEEE systems journal.

[210] Wang J. W., Wang H. F., Zhang W J, Ip W. H. and Furuta K. (2013b). Evacuation Planning Based on the Contraflow technique with Consideration of Evacuation Priorities and Traffic Setup Time, IEEE transactions on intelligent transportation systems, 14(1): 480-485.

[211] Wang, J.W., Wang H. F., Zhang, W. J., Ip W.H. and Furuta K. (2012b). On a Unified Definition of the Service System: What is its Identity?. IEEE Systems Journal, in press.

[212] Wang J. W. and Wang D. W. (2008). Particle swarm optimization with a leader and followers. Progress in Natural Science, 18(11): 1437-1443.

[213] Wolkerstorfer M., Statovci D. and NordstrÖm T. (2010). Robust Spectrum Management for DMT-Based Systems, IEEE TRANSACTIONS ON SIGNAL PROCESSING, 58(6): 3238-3250.

[214] Wolshon B., Urbina E., Levitan M. and Wilmot C. (2005). Review of policies and practices for hurricane evacuation. II: Traffic operations, management, and control, NATURAL HAZARDS REVIEW, 6(3): 143–161.

[215] Xie X. (2011). Service Quality Measurement from Customer Perception Based on Services Science, Management and Engineering, Systems engineering procedia,

1: 337-343.

[216] Xue D. and Dong Z. (2000). An intelligent contraflow control method for real-time optimal traffic scheduling using artificial neural network, fuzzy pattern recognition, and optimization., IEEE transactions on control systems technology, 8(1):183-191.

[217] Zhang W. J. (1994). An Integrated Environment for CADCAM of Mechanical Systems, Ph.D. thesis, printed by Delft University of Technology, The Netherlands, ISBN 90-370-0113-0, pp. 1-263.

[218] Zhang W. J. (2007). Is resilience the destiny for safety management paradigm? Presentation at the Northeastern University of China.

[219] Zhang W. J. (2008). Resilience Engineering. Presented at a seminar at Chinese Natural Science Foundation. http://homepage.usask.ca/~wjz485/Other%20Publication.htm. Download on Mar. 8, 2013.

[220] Zhang, W. J. and Lin Y. (2010). Principles of Design of Resilient Systems and its Application to Enterprise Information Systems. Enterprise Information Systems, 4(2): 99-110.

[221] Zhang W. J., Lin Y and Sinha N. (2005). On the Function-Behaviour-Structure Model for Design, Proceedings of the Canadian Engineering Education Association.

[222] Zhang W. J., Liu, X. and van Luttervelt C.A. (1997). On the Proposal of a New Theory, Methodology and Computer Aid for Virtual Enterprises Manufacturing Systems Design - in Relation to Partner Factories Selection, Proceedings of Int.

Conf. of World Manufacturing Congress' 97 (ISBN: 3-906454-08-8), New Zealand, 61-66.

[223] Zhang W. J., Werff van der K. and van Luttervelt C. A. (1993). A generic mechanism model for use in a CIM environment for the development of mechanized production machines, Annals of the CIRP, 42(1): 135-138.

[224] Zio E. (2009). Reliability engineering: old problems and new challenges, 94: 125-141.

Appendix A: FCBPSS Framework

It is first noted that the information of FCBPSS framework below is adopted from [Lin and Zhang 2004, Zhang et al. 2005].

The FCBPSS framework contains a set of core concepts, including: (1) structure, (2) state, (3) behaviour, (4) principle, (5) function, (6) context, (7) relationship among concepts (1)–(6), and (8) system decomposition. These concepts are further illustrated here. (1) Structure and state. A system has a structure that is a set of entities which are connected in a meaningful way. Entities are represented by a set of properties, and these properties are called the states. The states are given a name. The name of the state is the state variable. (2) Behaviour. The behaviour is the causal relationship or structure among a set of related state variables. (3) Principle. The principle governs or accounts for the behaviour in such a way that the causal relationship is derived from the principle. (4) Function and Context. The function is defined as a purpose in the mind of human users and can be realized by the system (structure) owing to certain behaviours existed in the structure. (5) System Decomposition. A system can be decomposed into subsystems and components within the system domain. The system structure, the behavior, the principle, the states, and the function concepts follow system decomposition. This means that it makes sense to speak of the behavior and states of a system (or subsystem, component). These concepts are related to each other. In particular, the structure concept is located at the bottom, followed by the state, the behavior, and the function concepts. The principle concept is situated between the state concept and behavior concept in order to give rationale for constraint equations such that given a set of values of the active state variables, the passive states are found through the evaluation of the constraint equations. The context concept is situated between the behavior and the function, which gives the rationale from the behavior to the function.

Appendix B: Permission to Reuse Figures

Fig. 2.4, Fig. 2.5, Fig. 2.6 and Fig. 7.2 Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.

2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.

3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]

2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.

3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go tohttp://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.